If it is determined in Step T66 by, for example, the **mobile** switching office, that the message transmitted from the **mobile** is not to be routed to an ADSL address, then it is determined by the **mobile** switching office whether the message is to be routed to an e-mail/internet address in Step T76. If the message is to be routed to an e-mail/internet address in Step T76, then the **mobile** switching office performs the upload operation to upload the message to an e-mail/internet address in Step T78. The **mobile** switching office optionally determines whether the upload was successful in Step T80, and if so, the process ends in Step T51.

If the **mobile** switching office optionally determines that the upload was not successful in Step T80, then the **mobile** switching office optionally determines whether the maximum number of attempts has been exceeded in Step T82, and if so, the process ends in Step T51. If the maximum number of attempts has not been exceeded in Step T82, then the message attempt counter is incremented in Step 874, and the process reverts to Step T78.

As discussed above, the present invention may also advantageously be used in the context of uploading a data, **voice mail** and/or electronic mail message to be transmitted to another user, **mobile** and/or land based, either terrestrial, **mobile** user, internet based and/or ADSL based user. One example of modifying the present invention to provide an additional upload feature is in accordance with U.S. Pat. No. 5,121,126 to Claggett, incorporated herein by reference. In the Claggett patent, users of telepoint services seeking to locate a base station will be in need of some means of location of the base station to permit the telepoint subscriber to approach sufficiently close to complete a call. The relevant telephone or base station is provided with a radio beacon capable of broadcasting simple pulse signals or more complex signals containing a variety of information messages. In simplest terms the base station will broadcast a signal which, in one way or another, says "Here I am."

The users of the new system and service may be provided with a highly portable receiver capable of being personally carried in the pocket or purse in the form of a credit card device, electronic calculator, electronic telephone directory device, watch, wristwatch, wristwatch band attachment, telephone paging receiver, **cellular** receiver, **mobile** receiver, or in or on an automobile. The radio beacon is arranged to provide the receiver with some type of indication of the approximate direction and distance or a city address or highway map location. The service may be arranged so that the transmitter responds to prompts or inquiries from the receiver.

A public pay phone or pay station in the PSTN is located along the road (highway 193) with a beacon transmitter or transceiver. The transmitter transmits a beacon signal having a predetermined range. The pay station may be

US-PAT-NO: 5819092
DOCUMENT-IDENTIFIER: US 5819092 A

TITLE: Online service development tool with fee setting capabilities
DATE-ISSUED: October 6, 1998

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	RULE 47
Ferguson; Charles H.	Cambridge	MA	N/A	N/A	N/A
Forgaard; Randy J.	Lexington	MA	N/A	N/A	N/A

ASSIGNEE INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	TYPE CODE
Vermeer Technologies, Inc.	Cambridge	MA	N/A	N/A	02

APPL-NO: 8/ 944365
DATE FILED: October 6, 1997

PARENT-CASE:
This application is a file wrapper continuation application, based on prior application Ser. No. 08/336,300, filed on Nov. 8, 1994, now abandoned, the benefit of the filing date of which is hereby claimed under 35 U.S.C. .sctn. 120.

INT-CL: [6] G06F015/00,G06F017/40

US-CL-ISSUED: 395/701,705/39
US-CL-CURRENT: 717/1,705/39
FIELD-OF-SEARCH: 395/701;705/39

REF-CITED:

U.S. PATENT DOCUMENTS

PAT-NO	ISSUE-DATE	PATENTEE-NAME	US-CL
5204897	April 1993	Wyman	380/4
5220501	June 1993	Lawlor et al.	705/43
5287270	February 1994	Hardy et al.	705/34
5325431	June 1994	Naruse	380/16
5359508	October 1994	Rossides	705/30
5367561	November 1994	Adler et al.	705/42

FOREIGN PATENT DOCUMENTS

COUNTRY	FOREIGN-PAT-NO	PUBN-DATE	US-CL

EP	0483576A3	May 1992
WO	WO 93/08661	April 1993

OTHER PUBLICATIONS

IBM Technical Disclosure Bulletin, "Multimedia Audio on Demand," vol. 37, No. 06B, Jun. 1994., pp. 451-460.

Telescan, Inc. New Release, Telescan Announces Major Online Technology Breakthrough, Jun. 1994, pp. 1-2.

Telescan, Inc. 1993 Summary Annual Report, pp. 1-13.

Ziff Announces Interchange Partners, The Seybold Report on Publishing Systems: The Latest Word, Apr. 1994, USA, p. 40.

Sherman, Sue, Spyglass Signs Agreement with NCSA to Enhance and Broadly Relicence . . . , Internet World, Jun., 1994, pp. 1-5.

Sherman, Sue, Frequently Asked Questions (FAQs) About Mosaic, Internet World, Jun., 1994, pp. 1-5.

Sherman, Sue, Passport to Electronic Publishing and Commerce on the Internet, Internet World, Jun., 1994, pp. 1-7.

Sherman, Sue, Spyglass, Inc., Spyglass Inc., Jun. 1994, pp. 1-2.

Mosaic: A Commercial World Wide Web Browser from Spyglass.RTM. (Promo Flyer), Internet Shopping Network, 1994, pp. 1-2.

SoftQuad, Inc., SoftQuad HoTMetal PRO (Promo flyer), 1994, pp. 1-5.

Glicksman, et al., Internet Publishing via the World Wide Web: Enterprise Integration Technologies, Proceedings of Groupware 1994, Aug. 1994.

InterServ: Desktop to Internet, WebMaster Services, Jun. 1994, pp. 1-6.

NovX Systems Integration, NovX Internet Security Solutions (Promo flyer), (undated).

NovX Systems Integration, Internet Integration Solutions, 1994, pp.1-2.

NovX Systems Integration, InterServ: Connecting the Global Network to your Desktop, 1994, pp. 1-2.

NovX Systems Integration, InterServ: NovX Product Brochure (Promo), 1994, pp. 1-4.

NovX Systems Integration, Internet Server Systems Spider-Series.TM., Jun. 1994, pp. 1-5.

NovX Systems Integration, Internet Server Systems Global-Series.TM., Jun. 1994, pp. 1-9.

Goodman, M., InterServ Offers WebMaster Internet Services for Online Multimedia Publishing (NovX Systems Integration), Jul. 1994, pp. 1-2.

Goodman, M., NovX Announces the "Spider-Series.TM.", a new line of prepackaged Internet servers, (Novx Systems Integration), Jul. 1994, pp. 1-2.

Goodman, M., NovX Announces the first commercially available line of Gopher, WWW, and FTP Internet servers and integration services (NovX Systems Integration), May 1994, pp. 1-3.

Mukhopadhyay, D., A Generic Information Retrieval System to Support Interoperability, (Bellcore), pp. 14-21.

Freund, R. et al., Special Issue on Heterogeneous Processing, Journal of Parallel and Distributed Computing, 1994, vols. 20-23, pp. 255-315.

Reddy, R., et al., Second Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, IEEE Computer Society Press, Apr. 1993, pp. 1-233.

Hurson, A.R., et al., Multidatabase Systems: An Advanced Solution for Global Information Sharing, IEEE Computer Society Press, 1994, pp. 1-59.

Zhao, J. et al., An Integrated Approach to Task-Oriented Database Retrieval Interfaces, British Computer Society, 1993, pp. 55-73.

Ligon III, W.B., et al., An Empirical Methodology for Exploring Reconfigurable Architectures, Journal of Parallel & Distributed Computing, 1993, pp. 323-337.

Graefe, G., et al., Encapsulation of Parallelism and Architecture-Independence in Extensible Database Query Execution, IEEE Transactions on Software Engineering, Aug. 1993, pp. 749-764.

Huang, Yen-Min, et al., Designing an Agent Synthesis System for Cross-RPC Communication, IEEE Transactions on Software Engineering, Mar. 1994, pp.

188-198.

Donaldson, Cameron, InQuisiX.TM. An Electronic Catalog for Software Reuse, Sigir Forum: A Publication of the Special Interest Group on Information Retrieval, Spring 1994, pp. 8-12.

Novel AppWare.TM.--A System for Developing Network Applications, White Paper, Novell, Inc., Jul. 1993, pp. 1-43.

Hughes, Kevin. Entering the World-Wide Web: A Guide to Cyberspace Siglink Newsletter, Mar. 1994, pp. 1-16.

The Annual Sigir '93: 16th International Conference on Research & Develop-,ment iin Information Retrieval, ACM Sigir Forum, Jun.-Jul. 1993, pp. 48-52.

Siglink Newsletter, vol. 2, No. 3, Dec. 1993, pp. 1-6.

Siglink Newsletter, vol. 2, No. 2, Sep. 1993, pp. 1-16.

Siglink Newsletter, vol. 2. No. 1, Mar. 1993, pp. 1-16.

Siglink Newsletter, vol. 1, No. 2, Dec. 1992, pp. 1-52.

Siglink Newsletter, vol. 1, No. 1, Mar. 1992, pp. 1-32.

Submission Guidelines, ITI--Software Patent Institute (SPI), Mar. 1993, pp. 35-45.

Hwang, Kai. Announcements of Special Issues, Journal of Parallel & Distributed Computing, 1994, pp. 24-39.

Garlan, D., et al., Special Issue on Software Architecture, IEEE Transactions on Software Engineering, Aug. 1994, p. 844.

Sigir '94: 17th Int'l. Conference on Research & Development in Information Retrieval--Call for Papers, Jul. 1994, p. 13.

ART-UNIT: 274

PRIMARY-EXAMINER: Voeltz; Emanuel Todd

ASSISTANT-EXAMINER: Chavis; John Q.

ATTY-AGENT-FIRM: Anderson; Ronald M.

ABSTRACT:
A visual editing system for creating commercial online computer services. The visual editing system creates online services that consist of a number of subservices. Each subservice is a program that provides a particular type of functionality to the online service. Different subservices exist for displaying hypermedia documents, searching directories and databases, displaying classified advertisements, providing a bulletin board system, etc. Each subservice has an associated database of information and a collection of scripts that handle events such as input from a user. The visual editing system of the present invention features a fee setting tool that allows the developer to develop a fee structure for an online service. The fee structure can handle both fees levied against users and third party content providers. For example, users can be levied fees for logging onto an online service, performing searches, or downloading information. Third party content providers can be levied fees for submitting advertisements or for executing a transaction with a user. Similarly, the fee setting tool also allows the developer to assign a payment system whereby users or content providers can be paid for certain actions. A user may be paid when that user that fills out a marketing questionnaire or wins a contest. A third party content provider can be paid when that third party content provider supplies valuable information desired by the users of the online service.

44 Claims, 26 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 25

BRIEF SUMMARY:

FIELD OF THE INVENTION

The present invention relates to the field of online computer services. In particular, the present invention discloses a software tool for setting fees in an online service, as part of a visually oriented tool for creating online services.

BACKGROUND OF THE INVENTION

With the increasing popularity of computer communications, many companies are becoming interested in advertising and supporting their products using an online computer service that can be accessed by customers. However, creating a large online computer service is an extensive task. To develop a sophisticated online service, such as America Online.RTM., CompuServe.RTM., Genie.RTM., or Prodigy.RTM., a company must have a large mainframe computer and customized software. Developing the customized software requires a competent programming staff and a good deal of time. Most companies do not have the resources

required to develop such systems, and thus cannot easily develop and maintain an online presence.

One way a company can contact millions of potential customers is to use the global Internet. The global Internet is a network of computer networks that links together millions of computer systems using the well defined TCP/IP protocol.

A new method of distributing and viewing information known as the World-Wide Web has recently become very popular on the global Internet. The World-Wide Web is a collection of servers connected to the Internet that provide multi-media information to users that request the information. The users access the information using client programs called "browsers" to display the multi-media information.

World-Wide Web servers store multi-media information in a document format known as HyperText **Markup** Language (HTML). The World-Wide Web servers distribute the HTML formatted documents using a specific communication protocol known as the HyperText Transfer Protocol (HTTP).

To access the multi-media information available on World-Wide Web servers, a user runs a client browser program that accesses the HTML formatted documents stored on the HTTP servers connected to the global Internet. The client browser program retrieves the formatted information and provides the information in an appropriate manner to the user. For example, the client browser program displays graphical image information as images on the user's graphical display screen; plays video information as video animation on the user's graphical display screen; displays text information as text on the user's screen; and plays sound samples using the speakers on the user's computer system. "Mosaic", one popular client browser program, is widely available to the users of the global Internet.

For a company that wishes to develop an online presence, creating a World-Wide Web Server would provide a feature rich online service available to customers and clients. A World-Wide Web Server can store images, text, animation, and sounds that provide information about the company. Furthermore, World-Wide Web Servers can be implemented on relatively simple computer systems, including personal computers.

Most World-Wide Web Servers are coupled to the global Internet. By deploying a World-Wide Web Server on the global Internet a company would create online service that is accessible to the millions of global Internet users.

Alternatively, a company can deploy a HTTP server that is available to customers through dial-up phone service. A dial-up HTTP server would be

accessible to customers and clients that do not have Internet access. Thus, by creating a simple HTTP server, any organization or corporation can create an online presence.

However, quickly creating the HTML formatted documents required for a World-Wide Web Server is not a trivial task. Moreover, the standard HTTP server software, without any additional programming, is very limited. For example, without custom extensions, an HTTP server cannot accommodate complex transactions between a user and the HTTP server or integrate a database system into an online service. Although it is possible to write custom extensions to the HTTP server software using a conventional programming language, such custom extensions are difficult to write except by experienced programmers. Thus, to be able to quickly deploy full-featured HTTP servers, it would be desirable to have a development tool usable by non-programmers that allows a developer to quickly and easily create a full-featured online service based upon the HTTP and HTML standards.

Many programming development tools are known in the art. These programming development tools range from tools which are developed and marketed as general purpose programming development tools to sophisticated special purpose development tools for developing specific types of applications.

For example, the Information Exchange Facility (IEF) general development tool, which is available from Texas Instruments, is used by professional programmers to develop application programs. Essentially, IEF provides a facility that allows a programmer to write "pseudo code" and IEF generates an intermediate source code program in a high level programming language (such as COBOL or C code) based on the "pseudo code". IEF is an example of what will be referred to herein as a "general purpose development tool" because it allows development of programs for essentially any purpose or application dependent on the input provided by the programmer.

In contrast to general purpose software development tools, many application programs themselves provide special purpose "development tool" capability. An example is the Paradox.TM. database program available from Borland International of Scotts Valley, Calif. The Paradox.TM. database allows end users to develop sophisticated database applications which would have been developed by professional programmers a few years ago. The Paradox.TM. database is but one example of a special purpose development tool.

Another example of a special purpose development tool, perhaps more pertinent to the present invention, is the Application Development Environment of Lotus Notes.TM. which is available from Lotus Development Corporation of Cambridge, Mass. The Application Development Environment of Lotus Notes provides features which are said to allow for rapid development of workgroup

applications such as sharing of documents between users over a network. Generally, Lotus Notes and, thus, its Application Development Environment, is directed at sharing of documents among persons in an authorized work group. For example, a Lotus Notes application can be envisioned which would allow for sharing of key patent applications among patent examiners in a particular art group at the United States Patent Office.

The Lotus Notes Application Development Environment provides for such features as (i) application design templates which are said to allow sophisticated applications to be built by customizing pre-built applications such as document libraries, form-based approval systems, project tracking applications and status reporting systems; (ii) security; (iii) database access; and (iv) discussion groups. However, while these features are useful, the Lotus Notes Application Development Environment, as well as Lotus Notes itself, has its shortcomings as admitted to by even Lotus Development Corporation itself:

Lotus Notes was not intended to be used as a transaction-processing front-end to an operational database system. Operational systems are those which support transactions that are essential to the operation of an organization. Examples of these systems would be traditional order entry . . . Lotus Notes: An Overview, October, 1993, pg. 11

It has been recognized by the present invention that many of these functions neglected by Lotus Notes are very important when developing publicly accessible online systems. Specifically, the ability to perform commercial transactions that involve order entry systems would allow an online system to sell goods and services to computer users. It is now recognized by the present invention that many functions such as traditional order entry systems and the like will someday be carried out over computer networks by allowing a customer to place orders for goods and services directly with an online service. By way of example, even today, food orders can be placed with restaurants over computer networks; videos can be reserved at the local video store; and banking transactions can be carried out simply by logging onto a computer network.

Four different types of commercial transactions might commonly occur in a commercial online service. First, a user may be charged for the right to access all or parts of a useful publicly accessible online system. Second, the online service may pay the user for performing some type of action such as winning a contest or completing a marketing survey. Third, an online service may charge a content provider for placing certain information on the online service. For example, a content provider can be charged for placing an advertisement on the online service. Finally, a content provider can be paid by the online service for providing information that users may wish to access, can be can be provided on a for-fee basis. Conversely, an online service

provider may wish to pay third party content providers for placing useful material on the online service.

Thus, when creating a publicly accessible online system, it is desirable to include the ability to define fee structures for accessing parts of the online system and/or ordering other goods or services. However, creating a sophisticated commercial online service with such features usually requires specialized programming.

The ability to set fees to be paid by the user for an amount of data accessed, the time spent "logged on" to the online service, or the purchase of particular merchandise is one example of distinction from Lotus Notes. Lotus Notes is not only admitted (by even Lotus Development Corporation) as lacking transaction oriented capability as may be required by such applications, but it also does not provide the metering functions to keep track of the information necessary to assign such fees as is required by these applications. As such, the video store, restaurant or bank (by way of example) is left with the need to employ professional programmers for their individual applications.

Thus, it has been discovered that there exists a need to create online system development tools that include features, functions and capabilities to support commercial online services such as the aforementioned fee setting function.

These and other aspects of the present invention will be described in greater detail with reference to the below detailed description and the accompanying figures.

SUMMARY AND OBJECTS OF THE INVENTION

It is thus an object of the present invention to provide a fast, user-friendly method of designing and deploying an online system.

It is a further object of the present invention to provide a visual editor that allows a developer to easily create distributed online services. In particular, it is an object of the present invention to allow a developer to create customized HTTP server software and accompanying HTML documents in order to deploy a World-Wide Web Server.

It is yet a further object of the present invention to provide a sophisticated fee setting tool that allows a developer to assign a system of fees for access to an online service. The fee setting tool allows complex fee arrangements to be created using a well defined scripting language.

These and other objects are provided by the Online Designer of the present

invention. The Online Designer is a visual editor that allows a developer to create an online service that consists of a set of standardized subservices. The subservices include a Hyperdocument/Commerce subservice for displaying hyperdocuments and performing electronic transactions, a Classified Advertisement subservice for implementing electronic classified advertisements, a Reference subservice for implementing online reference works, a Directory Lookup subservice for implementing online searchable directories of information, a Bulletin Board subservice for providing a means for allowing users to post and view messages, a Document Retrieval subservice to provides a means for retrieving documents, an Electronic Publishing subservice that provides electronic editions of newspapers or magazines that may be downloaded, and a Meta-Service subservice that provides access to other external online services.

The visual editing system of the present invention features a unique fee setting tool that allows a developer of an online service to develop a fee structure for the online service. The fee structure for the online service can handle fees levied against both users and third party content providers. For example, a user can be levied fees for logging onto an online service, performing searches, or downloading information. Third party content providers can be levied fees for submitting advertisements or for executing a transaction with a user. Similarly, the fee setting tool also allows the online service developer to assign a payment system whereby users or content providers can be paid for certain actions. For example, a user may be paid when that user fills out a marketing questionnaire or wins a contest. A third party content providers may be paid when that content provider supplies valuable information desired by users of the online service.

Other objects, features and advantages of the present invention will be apparent from the accompanying drawings, and from the detailed description that follows below.

DRAWING DESCRIPTION:

BRIEF DESCRIPTION OF THE DRAWINGS

The objects, features, and advantages of the present invention will be apparent from the following detailed description of the preferred embodiment of the invention with references to the following drawings.

FIG. 1 illustrates a block diagram overview of an online service that is implemented with the Molisa platform.

FIG. 2 lists the general steps for an electronic commerce transaction with an online service.

FIG. 3a illustrates how the Online Designer is used to create an online service.

FIG. 3b illustrates how a hyperdocument is edited using the Online Designer.

FIG. 4 lists the available subservice design programs.

FIG. 5 illustrates a hyperlink from a Reference subservice to an order form in a Hyperdocument/Commerce subservice.

FIG. 6 illustrates a hyperlink from a Directory Lookup subservice to an rental form in a Hyperdocument/Commerce subservice.

FIG. 7 illustrates the set of Utility Subtools in the Online Designer development tool.

FIG. 8 illustrates a block diagram example of an online service.

FIG. 9 illustrates a service gallery containing online services that may be edited.

FIG. 10 illustrates a Connectivity View of the online service of FIG. 8.

FIG. 11 illustrates a WYSIWYG view of a hypermedia document.

FIG. 12 illustrates a screen display of the Hyperdocument Designer Script View.

FIG. 13 illustrates a hyperlink view of a hyperdocument.

FIG. 14 illustrates a block diagram of the views supported by the Hyperdocument Designer.

FIG. 15 illustrates a screen display of a hypermedia document.

FIG. 16 illustrates a screen display of a hypermedia document used to order a product.

FIG. 17 lists the different views provided by the Lookup Designer subtool.

FIG. 18 illustrates a submit form in the Form View of the Lookup Designer subtool.

FIG. 19 illustrates a view form in the Form View of the Lookup Designer subtool.

FIG. 20 illustrates a query form in the Form View of the Lookup Designer subtool.

FIG. 21a illustrates a first screen display of the Script Editor.

FIG. 21b illustrates a second screen display of the Script Editor.

FIG. 22 illustrates an SQL query between an online subservice and an SQL database.

FIG. 23 illustrates a screen display of the Fee Setting subtool.

FIG. 24 illustrates a screen display of the Fee Specifier editor subtool.

DETAILED DESCRIPTION:

NOTATION AND NOMENCLATURE

The detailed descriptions which follow are presented largely in terms of algorithms and symbolic representations of operations within a computer system. These algorithmic descriptions and representations are the means used by those skilled in the data processing arts to convey the substance of their work most effectively to others skilled in the art.

Generally, and within the context of this application, an algorithm is conceived to be a self-consistent sequence of steps leading to a desired result. These steps are those requiring physical manipulations of physical quantities. Usually, though not necessarily, these quantities take the form of electrical or magnetic signals capable of being stored, transferred, combined, compared, and otherwise manipulated. It proves convenient at times, principally for reasons of common usage, to refer to these signals as bits, values, elements, symbols, characters, terms, numbers, or the like. It should be borne in mind, however, that all of these and similar terms are to be associated with the appropriate physical quantities and are merely convenient labels applied to these quantities.

Further, the manipulations performed are often referred to in terms, such as adding or comparing, which are commonly associated with mental operations performed by a human operator. No such capability of a human operator is necessary, or desirable in most cases, in any of the operations described herein which form part of the present invention; the operations are machine operations. Useful machines for performing the operations of the present invention include general purpose digital computers or other similar devices. In all cases, a distinction is maintained between the method of operations in operating a computer and the method of computation itself. The present

invention relates to method steps for operating a computer in processing electrical or other physical signals (e.g., mechanical, chemical) to generate other desired physical signals.

The present invention also relates to apparatus for performing these operations. This apparatus may be specially constructed for the required purposes, or it may comprise a general purpose computer as selectively activated or reconfigured by a computer program stored in the computer. The algorithms presented herein are not inherently related to a particular computer or other apparatus. In particular, various general purpose machines may be used with programs written in accordance with the teachings herein, or it may prove more convenient to construct more specialized apparatus to perform the required method steps. The required structure for a variety of these machines will appear from the following description.

DETAILED DESCRIPTION OF AN EMBODIMENT OF THE PRESENT INVENTION

Methods and apparatus for implementing a development tool for creating online services are disclosed. In the following description, for purposes of explanation, specific nomenclature is set forth to provide a thorough understanding of the present invention. However, it will be apparent to one skilled in the art that these specific details are not required to practice the present invention. For example, the present invention is disclosed with specific reference to the HyperText **Markup** Language (HTML) and the HyperText Transfer Protocol (HTTP). However, the teachings of the present invention can easily be used with other hypertext document formats and other transport protocols.

Overview

The present invention provides a visually oriented software development tool for the design, construction and modification of online computer services. The development tool of the present invention allows a user to create online services using existing information sources such as databases, files, and applications that are external to the online service itself. An online computer service created with the development tool can offer the following options:

Search, view and edit information

Download, print or file information

Enable the information for commerce

Control access to the information

Examples of types of online services that can be built with development tool of the present invention include document viewing services, electronic commerce services, directory lookup services, classified advertisement services, reference services, electronic bulletin board systems, document retrieval services, electronic publishing services, an electronic service store for purchasing online services, and a global service-of-services that is used to locate and connect to other online services. To create commercial online services, the development tool of the present invention includes a sophisticated fee setting tool that levies of pays fees to users and content providers under defined conditions.

The online service development tool of the present invention is one part of a comprehensive architected platform for deploying distributed online services. The online services created with the development tool of the present invention utilize standardized Application Program Interfaces (API's) for communication between the various components. The overall platform architecture is referred to as the Modular Online Information Services Architecture, or Molisa. Molisa includes client software, server software, administrative software for recording and analyzing online service usage, and the online service development tool described in this document. The software components of the Molisa platform are hardware independent, and thus can be implemented on several different computer architectures.

The invention's design characteristics are described here in the context of its preferred embodiment, a development tool for the Molisa online services platform. The Molisa platform leverages existing HyperText Transfer Protocol (HTTP) based World-Wide Web servers, and Mosaic and other HTTP client browsers (with software extensions), on the global Internet. However, the design principles of the present invention are largely applicable to online services in other settings, including non-architected centralized online services, other decentralized online services, and services in which the client and server software reside on a single machine (such as CD-ROM based information services).

The Application Program Interfaces that define communication between the client software and server software are largely independent of the underlying transport protocol. For example, the development tool of the present invention does not require that the client and server computers communicate using HTTP or the underlying TCP/IP protocol. Any suitable transport protocol, across Local Area Networks (LAN's), Wide Area Networks (WAN's), dial-up or leased telephone lines, etc., may be used between the client hardware and the server hardware.

FIG. 1 illustrates a block diagram overview of an online service being used by three users that is implemented with the Molisa platform. A server hardware

platform 100 comprises a general purpose computer system coupled to a communications network 150. The HTTP server software 101 and HTTP extension software 103 run on the server hardware platform 100. The HTTP server software 101 drives the online service using information stored within the service repository 107. The HTTP extension software 103 provides additional functionality for the online service that is not available in standard HTTP server software. For example, the HTTP extension software 103 might access a back end database.

An online service development tool 109 is used to create the data structures, documents, and scripts that are stored in the server service repository 107 and supply the HTTP extension software 103. The HTTP server software 101 accesses the data structures, documents, and scripts stored in the service repository 107 to implement an online service. Software for the development tool 109 is usually located on a development computer system that is coupled to the server system across a communications network 150 as illustrated in FIG. 1. Alternatively, software for the development tool 109 may run on the actual server computer system.

Each user accesses an online service created with the development tool 109 using compatible client software. In FIG. 1, three client hardware platforms: Windows.RTM. platform 160, Macintosh.RTM. platform 170, and UNIX.RTM./X-Windows platform 180 are illustrated. Each different client hardware platform must have a copy of client browser software that is compatible with the HTTP server software 101 and the information stored within the service repository 107. Each client hardware platform may also have a local service repository. The local service repository at each client hardware platform contains information that is available locally to the user of the specific client hardware platform. The local service repository can also act as a cache to store information retrieved from the main service repository 107.

The communications network 150 couples the users running client software with the online service server software running on the server hardware. In the present embodiment, the communications network 150 is a packet switched network implemented using TCP/IP protocol. However, the communications network 150 could simply be the existing telephone network.

Using the Molisa platform as illustrated in FIG. 1, small and large service providers can run an online service using existing heterogeneous computer equipment and existing data in its original native form and location. Since the Molisa platform uses standardized well-defined Application Program Interfaces (API's), third parties can develop enhancements, extensions, or replacements for the client software, the server software, the metering software, or the online service development tool software.

Furthermore, the online service development tool software of the present invention is divided into several different subtools that each have well defined subtool Application Program Interfaces (API's). By having well defined subtool API's, third parties may create improved subtools to replace the original subtools. Alternatively, new subtools can be added to the development tool software to handle unforeseen development.

Electronic Commerce

The Molisa platform places a particular emphasis on commerce-enabling any information source that is electronically accessible. The Molisa platform uses the general steps illustrated in FIG. 2 for electronic commerce in an online service. Initially, the service user views general online information about goods or services that are external to the service, as stated in step 210. This is usually done using a hypermedia document that contains images and text describing the goods and services. Next, the user initiates an electronic transaction to download, price, purchase, rent, reserve, etc. the online hyperdocument itself or the goods/services that the hypermedia document information describes, as stated in step 220. In response to the user's action, the online service processes the electronic transaction initiated by the user, as stated in step 230. Using a Fee Computation defined in the Computation Language of the present invention, the online service may charge or pay a user or content provider as stated in step 240. Finally, the user views the results of the electronic transaction, by viewing the downloaded information, or by viewing a confirmation of the electronic transaction involving goods or services, as stated in step 250.

In the transaction model of FIG. 2, the notion of "transaction" can take several forms, and the development tool invention supports each of the following:

Real-time electronic transaction: A transaction can debit a user's account, check and subtract from inventory, mark an item as reserved, reference up-to-date online information, etc., all by immediately accessing the electronic databases that contain the relevant data. The invention supports such real-time transactions with a Script Language that provides direct access to any electronic databases that are accessible from the server.

Real-time manual transaction: For manual systems (e.g., a clerk checks inventory by looking in the back room, and then responds to the user in real-time), or electronic systems that are not accessible from the server computer, human intervention might be required to complete a transaction. The invention supports these transactions with Script Language primitives that allow for real-time cooperative activity between users and a representative of the online service provider.

Delayed electronic transaction: In certain cases, an online service may wish to queue a series of transactions for later batch processing. For example, an online service could queue all the transactions for a particular day and transmit all the transactions for that day during the night to save on long-distance telephone charges when dialing-up a remote computer. Alternatively, an online service may wish to issue a transaction against a computer that provides only electronic mail access. To support these delayed electronic transactions, the invention includes Script Language primitives that: (1) perform file input/output (to queue transaction requests), and (2) send/receive electronic mail to automatic agents on other computers.

Delayed manual transaction: Some online services can require manual transactions that do not occur in real-time. For example, an online service run by an antique dealer can allow users to submit bids for items advertised on the service, and the antique dealer can consider all received bids at the end of the business day. To support these transactions, the invention's Script Language includes primitives that can submit and receive electronic mail between the user and a representative of the service provider.

The use of examples will best illustrate how the development tool can be used to commerce-enable existing sources of electronic information. For example, the development tool can convert the digital source information used to create a printed catalog into a commerce enabled online service. The created online service displays the contents of the catalog on a user's display screen. A user can check the available stock and place an order for any item in the catalog. Also, for example, the development tool can convert a list of classified advertisements into an online service where advertised goods may be electronically reserved with a deposit or purchased outright. To update the electronic list of classified advertisements, users may electronically submit new advertisements to the online service for a set fee.

Another type of online service the development tool can create is a service that selects specific items from a collection of newsfeeds, based on a user's previously registered interests, and assembles a customized electronic newspaper for which the user is charged a fee. Payment for any transaction with any online service can be handled using secure, **authenticated** electronic transaction techniques as is well known in the art. Alternatively, other methods of payment such as credit card payment, electronic funds transfer, or external payment mechanisms (e.g., mailing a check) can be used.

To create online systems that are prepared for commerce, the online system development tool includes a Fee Setter for assigning fees and a sophisticated Script Language for creating scripts that control commerce transactions. The online system development tool of the invention embodiment is referred to as

the Online Designer. The Online Designer is a visual editing system that allows a developer to create online services using graphical screen displays and cursor control device such as a mouse. The Online Designer is composed of several distinct, cooperating, visually compatible subtools. Although some of the Online Designer subtools are implemented as separate programs, all of the Online Designer subtools appear to the user as an integral part of the Online Designer, and are described here as such.

The Online Designer online system development tool can be used to create sophisticated, yet easy-to-use online services. Some of the features of an online service designed using the Online Designer development tool include:

Display of "hypermedia" documents: Hypermedia documents present text, images, video, and/or sound to a user of the online service. Hypermedia documents may function as on-screen input forms by including visual objects for user input: text fields, checkboxes, option buttons, command buttons, and drop-down list boxes. In the present embodiment, the hypermedia document format supported by Online Designer is the HyperText **Markup** Language (HTML). HTML is the HyperText format supported by HTTP servers comprising the World-Wide Web (WWW) on the global Internet.

Display of portable documents: Portable documents preserve the exact printed appearance of a document (fonts, illustrations, etc.), and can be viewed on different hardware and software platforms. A portable document can be generated by any software application that supports printing. A specially designed print driver converts the printer commands into the portable document format. Examples of portable document formats include Acrobat.RTM. by Adobe of Mountain View, Calif. and WordPerfect.RTM. Envoy by WordPerfect Corporation of Orem, Utah. The portable document may be viewed on a workstation display screen as part of an online service. Collectively, hypermedia documents and portable documents are referred to in this document as "hyperdocuments."

Support for "hyperlinks": Hyperlinks are visual buttons, images, or highlighted text that are associated with other documents, images, sound clips, video clips, or other online services. To move to the associated object, a user selects a "hotspot" with a cursor control device or chooses the hyperlink with the computer keyboard. Hyperlinks appear within hyperdocuments.

Support for full-text index/search/retrieval: Allows for quick search through large collections of online documents. The user can specify the search criteria using an appropriately designed hypermedia input form.

Attribute-based searching: A user may search through documents by specifying various document attributes such as the date of the last update, the size of

the document, the size of the fee for downloading the document, etc.

Downloading data or programs: Allows data or programs to be downloaded from the online service to the local client computer system. Downloaded data or programs can later be executed, viewed, printed, or filed at the local client system.

Support for communication between different online services: Service-to-Service Protocol is a communication protocol whereby different online services can communicate information. Using the Service-to-Service protocol of the present invention, an online service can: (1) transfer control to another online service; (2) act on behalf of the user to query or update another online service; (3) automatically update another online service without user initiation; (4) appear to be seamlessly part of another online service; (5) keep a record of how many times users traverse to another online service; (6) pass along automatic user registration data to another online service; (7) automatically register a new online service with a service-of-services or "yellow pages" service; (8) check whether another server is running a particular online service or type of service; and (9) exchange usage and metering information, for aggregation and later analysis.

Support for an Online Designer Script Language: The Online Designer supports two different types of scripts: Event Scripts and Function Scripts. An Event Script is associated with a particular event for a particular visual object in the online service. For example, there could be an Event Script associated with a "Mouse Down" event for a "Search" button on a "ListingQueryForm" hypermedia document in an electronic "white pages" service. The event script associated with the mouse-down event would specify how to convert the user input fields on the "ListingQueryForm" form into a query for the text search/retrieval engine on the server. In general, there can be several Event Scripts associated with a single visual object, potentially one script for each type of event that is defined for that visual object. A Function Script contains a single named function (subroutine) that can be shared and invoked by multiple other scripts in the same service.

Launching and control of other software applications: These capabilities are achieved using inter-application communication techniques such as Windows DDE, Windows OLE, OpenDoc, keystroke stuffing, terminal emulation, command-line invocation, batch file invocation, and the like. For example, an online service can compute the quantity discount for a catalog item by automatically launching a spreadsheet program, plugging the item number and quantity into certain prearranged spreadsheet cells, invoking a spreadsheet macro to compute the discount, and obtaining the item price from a prearranged result cell. Other examples include launching and controlling applications for payroll, inventory, purchasing, and Manufacturing Resources Planning (MRP).

Directly and transparently accessing real-time data sources: Structured Query Language (SQL), Open Database Connectivity (ODBC), and other published and proprietary data access methods can be used to access real time data sources. For example, a catalog shopping online service can check the available stock on a certain merchandise item by issuing an appropriate SQL query to the inventory database. The inventory database would return the information to the online service software such that the online service could provide the information to the user or perform an electronic transaction.

Accessing and manipulating control equipment: Equipment such as heating/ventilation/air-conditioning systems, security systems, and lighting can be accessed and controlled.

Replication of online service content: The service's content and structure can be replicated to other online services on-demand or on an automatic, regularly scheduled basis.

Metering of user usage patterns for the online service: This can include the number of users who access the service, the duration of each user's connection time, the number of times that a certain part of the service is accessed, the number of times that a user was "referred" to this service by hyperlinking from another service, etc. This data can be used to levy fees for users, advertisers, or information providers, or to tune the service itself.

Controlling access to information: The available information on an online service can be controlled utilizing passwords, encryption, and assigning specific access rights to specific users.

Real-time cooperative activity: Support of real-time cooperative activity between two or more users, or between users and a representative of the online service provider. For example, a multi-person game between users, or a user entering an online query and receiving a real-time response from a service representative.

Capturing and Editing Images: Allowing a user or service operator to capture an image to be displayed as part of an online service (for example, a logo for a "yellow pages" directory listing, or a photograph to accompany an online classified advertisement), by faxing an image directly to the server, sending an image to the server using electronic mail, or scanning an image at the client workstation and electronically transmitting the image to the server. If a user does not have access to facsimile or scanning equipment, the user may physically send or deliver the photograph or graphic to a service operator, who will electronically capture the image on behalf of the user and transmit the image to the online service server.

Building an electronic service store: Allows a user to download entire online services (the structure and/or content), usually for a fee. The user can then deploy those services on the user's own computer equipment.

Searching and connecting to other online services: Allows a user to access a service-of-services which will search and connect to other online services.

The Online Designer Subservices

Using the Online Designer, a user (referred to here as the online service "developer") can develop one or more online services using a graphical editor. Each online service consists of one or more types of "subservices." Each subservice is a server program for handling a particular type of online service user interaction. Each subservice program has an associated database that stores the information that can be provided to the user and a set of scripts for handling events.

FIGS. 3a and 3b illustrate how the Online Designer is used to create a set of subservices that comprise an online service. First, at step 310 user decides if another subservice should be created or edited, if not, then the online service is complete. When creating a new subservice, the developer may retrieve an existing sample subservice to start from as stated at step 315. At steps 320 and 325, the developer loads in an existing subservice document. If no document (database) exists for the new subservice, the developer can import an existing document. If the developer wants to preserve the printed appearance of the existing document, the Portable Document Converter is used as stated at step 335, otherwise the Hypermedia Document Converter is used at step 337. The developer then edits the subservice by editing the associated document (database) at step 340.

FIG. 3b illustrates, in detail, how a document can be edited using the Online Designer. The Hypermedia Editor is used to modify the appearance of the document at step 342. The Hypermedia Editor cannot be used if the document is a portable document. Any inline images can be edited using third party image design tools such as paint programs at step 344. The interactive elements of a subservice are edited by creating input forms with the Hypermedia Editor at step 346, and creating event scripts that process the input using the script editor at step 347.

Referring back to FIG. 3a, the hyperlinks within a document and to other online services or subservices can be created and edited using the Hyperlink Editor at step 351. Finally, the developer can save the created subservice for the online service at step 362.

The Online Designer includes specific Designer Subtools for designing each different type of subservice. In general, an online service may include more than one subservice of the same type or of different types. The following nine types of subservices are examples of subservices supported by the Online Designer: Hyperdocument/Commerce, Directory Lookup, Classified Advertisement, Reference, Bulletin Board, Document Retrieval, Electronic Publishing, and Meta-Service. Additional types of subservices can be added later.

FIG. 4 illustrates a block diagram of all the subservice designer tools of the Online Designer. More subservice design tools can be added in the future. As illustrated in FIG. 4, the Lookup Designer 414 is used to design Directory Lookup subservices, Classified Advertisement subservices, and Reference subservices since these three types of subservices share many similarities.

Hyperdocument/Commerce Subservice

Referring to FIG. 4, the Hyperdocument Designer 412 is used to design Hyperdocument/Commerce subservices. A Hyperdocument/Commerce subservice displays hypermedia information to a user of an online service. A Hyperdocument/Commerce subservice can optionally allow a user to purchase goods or services described by the displayed hypermedia information. Hyperlinks and full-text searches allow the user to move through the different hyperdocuments that comprise a Hyperdocument/Commerce subservice.

An example of a Hyperdocument/Commerce subservice is an electronic shopping system, where the user views an online catalog of goods or services and potentially submits an electronic order for goods. The user's electronic order is processed by an event script associated with the Hyperdocument/Commerce subservice. The Hyperdocument/Commerce subservice can also be used to display online documentation or help.

Directory Lookup Subservice

Referring to FIG. 4, the Lookup Designer 414 is used to design Directory Lookup subservices. A Directory Lookup subservice provides an online, searchable directory of information. For example, a Directory Lookup subservice can store a directory of persons, companies, or other entities. Each entry in the directory can list a name, an address, and any other related information; essentially, an online implementation of telephone "white pages" listings. Alternatively, the entries in a directory can be hyperdocuments that include company descriptions and advertisements, and can be arranged in categories, much like conventional telephone book "yellow pages" listings. In either case, entries are searchable by name, by category, or using full-text search techniques with user-specified keywords.

Each directory entry can optionally provide hyperlinks to other entries. Furthermore, each entry can contain a hyperlink to a dedicated online service that provides additional information about the entry. Using the directory subservice, qualified users can submit new entries, which immediately become available for retrieval in subsequent searches of the directory subservice by other users.

Classified Advertisement Subservice

Referring to FIG. 4, the Lookup Designer 414 is also used to design Classified Advertisement subservices. A Classified Advertisement subservice implements an online version of classified advertisements. Users can search existing classified advertisement listings. Classified advertisement submissions are searchable by category, geographical area, the name of the submitter, or using full-text search techniques with user-specified keywords. Furthermore, end users can submit new classified advertisement listings of their own. The online service can charge a fee for submitting a new classified advertisement.

Reference Subservice

Referring to FIG. 4, the Lookup Designer 414 is also used to design Reference subservices. A Reference subservice implements an online reference work, such as a dictionary, thesaurus, or encyclopedia. A user can search the contents of a Reference subservice by the name of the entry, or using full-text search techniques with user-specified keywords. The online service provider controls the content of a Reference subservice. However, the online service allows the user to submit additional personal entries that are seamlessly integrated into the subservice, but are only seen by that user. These personal entries can be stored within the service repository of the client hardware system.

Bulletin Board Subservice

Referring to FIG. 4, the Bulletin Board Designer 416 is used to design Bulletin Board subservices. A Bulletin Board subservice provides a means for allowing users to post and view messages about a particular topic. Users may read and search through the existing messages. A user may reply to an existing message, or reply to a reply, thus creating a "conversation thread" from an original message.

The messages are divided into different sections. Messages within a given section conform to a submission form designed for that section by the online service developer. The submission form can contain various data fields that are specific to that message section, in addition to a text input area on the

form for typing the message to be posted. The developer also specifies which message data fields should appear in the summary view for each bulletin board section. The summary view lists the relevant header information for the messages in that section.

Document Retrieval Subservice

Referring to FIG. 4, the Retrieval Designer 418 is used to design Document Retrieval subservices. A document Retrieval subservice provides a means for users to retrieve documents and other files such as word processing documents, spreadsheets, text files, databases, images, sound files, video files, executables, etc. Users can find such documents using full-text search techniques with user-specified keywords, for those document types that can be viewed as text. Users can also find documents by category and subcategory, when hierarchical browsing is supported by the online service. A document Retrieval service can be used to provide searchable access to a large corpus of text, to make files on a file server available to geographically remote offices within a company, or to provide software updates to customers.

Electronic Publishing Subservice

Referring to FIG. 4, the Publishing Designer 420 is used to design electronic Publishing subservices. An Electronic Publishing subservice provides a user with an electronic edition of a newspaper or magazine that the user may download to the user's local client hardware. An Electronic Publishing subservice can create a customized daily newspaper, that provides only news stories that match certain criteria provided previously by the user. Downloaded material may take the form of static documents, or hypermedia documents with images, sound, video, and hyperlinks to move through the hypermedia document.

Meta-Service Subservice

Referring to FIG. 4, the Meta-Service Designer 422 is used to design Meta-Service subservices. A Meta-Service subservice provides a service-of-services that is designed specifically to help a user find another online service. Using the Meta-Service subservice, a user can look for another online service, using keyword searches, categories, alphabetic listings, etc. An online service listing can include a description of the online service, the categories to which the online service belongs, and a visual icon for the online service. Once the user finds a desired online service, the meta-service provides a direct connection to the online service. A Meta-Service subservice can itself lead to other meta-service subservices, in a hierarchical or network fashion.

Hyperdocument/Commerce Subservice

In a common use for the invention, a Hyperdocument/Commerce subservice will be combined with other subservices to commerce-enable those other subservices.

For example, an online service that includes a Reference subservice that allows the user to peruse the current Books In Print can also include a Hyperdocument/Commerce subservice to enable books to be reserved or purchased. This is illustrated in FIG. 5, in which a Reference subservice containing book descriptions has a particular book reference with a hyperlink. When viewing a particular book entry in the Reference subservice, the user clicks a button on that book entry to transfer directly to the Hyperdocument/Commerce subservice. The Hyperdocument/Commerce subservice will be invoked to display an on-screen order form so the end user can buy the book. The user types the necessary ordering information into the order form and transmits the information. The Hyperdocument/Commerce subservice contains a script for processing the information entered on the on-screen order form.

A second electronic commerce example is provided with reference to FIG. 6. FIG. 6 illustrates an automobile rental agent's use of the Online Designer to create and deploy an online service that includes a Directory Lookup subservice that displays information about the available cars to rent. If a user selects a particular car from the Directory Lookup subservice, then the online service transfers control to a Hyperdocument/Commerce subservice that displays an automobile rental form to the user. To rent the car, the user fills in the on-screen form. The information entered into the on-screen form is processed by a script in the subservice electronically transmitted to the automobile rental agency.

The Online Designer makes a distinction between the "framework", the "structure", and the "content" of an online service. The "framework" is the architected online services platform Molisa for which Online Designer develops online services. The Molisa framework provides the domain independent infrastructure and includes the Online Designer, the subservice programs created with the Online Designer, the documents created with the Online Designer for use by the subservices, and the client software used to access the online service created with the Online Designer.

The "structure" of an online service is composed of those portions of the service that define its behavior. Classified advertisement classifications, a bulletin board submission form, and hyperlink attributes are examples of the components that comprise the structure of an online service. The structure includes the selected subservices and how the selected subservices are connected together.

The "content" of an online service is the information that the each of the subservices delivers to users. Some of the content of an online service can be static, provided by the developer when the online service is designed; and some of the content can be dynamic, provided by the developer or other users at run-time without requiring further online service design work. Examples of static content includes the screen displays for different regions of an online service. Examples of dynamic content includes bulletin board messages written by users and classified advertisements submitted by users.

Hyperdocuments are sometimes part of the structure of an online service, and sometimes part of the content. For the purposes of Online Designer, a hyperdocument is considered part of the service's structure if it is in a Hyperdocument/Commerce subservice, or if it is a form. Otherwise, the hyperdocument is considered a part of the service's content if the hyperdocument is displayed in any other type of subservice.

Using the Online Designer, a developer can create "templates" for the structure and/or content of an online service, in addition to developing entire online services. A "structure template" is a partially developed online service structure, whose development can be easily completed by developers who provide the missing details to create a fully functional and customized online service. Similarly, a "content template" is a partially developed content module for an online service. Templates can be separately packaged and provided to other developers to expedite the development of online services using Online Designer.

The structure and content of an online service are stored in the Service Repository. The Service Repository is a database system that is potentially distributed between the client workstation and one or more servers, depending on the design of the particular online service. When the data is distributed, it still appears to the developer and user as a seamless whole. Each Online Designer subtool stores the data for its subservice in the Service Repository, and the various Utility Subtools access data from the repository for a single subservice or multiple subservices.

The Online Designer includes replication support for the Service Repository itself. This is useful in cases where the developer's workstation does not have full-time direct access to the Service Repository for a certain online service. For example, this can occur if the service was originally developed by another person. In such cases, Online Designer can replicate the components of the online service from the server where the service resides down to the developer's workstation, and then replicate the components back to the server when the modifications are complete.

There can also be more than one developer who regularly maintains an online

service. In such cases, it becomes important to prevent multiple developers from modifying the same part of the service simultaneously, because one developer's work will overwrite another when the changes are replicated back to the server. To address this issue, Online Designer supports version control techniques, as known in the art, allowing a developer to "check out" a service component for modification. When checked-out, the component may be viewed but not modified by any other developer until the component is later "checked-in" by the developer who checked it out.

Online Designer Utility Subtools

The Online Designer tool provides the organizational structure for developing online services. It also provides access to the various subtools that allow the developer to design the details of online services. Online Designer includes specific Designer Subtools for each type of subservice. The various Utility Subtools are accessible directly from the Online Designer tool, as well as from those Designer Subtools that apply.

FIG. 7 illustrates a block diagram of the Utility Subtools. The list of Utility Subtools that support Online Designer and the Designer Subtools include a Hypermedia Editor 711, a Portable Document Editor 712, a Hyperlink Editor 714, a Hypermedia Document Converter 716, a Document Harvester 718, a Script Editor 720, a Fee Setter 722, a Replicator 724, a Metering Tool 726, a Repository Browser 728, a Debugger 730, and a Help Editor 732. A description of each utility subtool is hereby provided.

Hypermedia Editor

The Hypermedia Editor 711 is a What-You-See-Is-What-You-Get (WYSIWYG) visual editor for creating and editing hypermedia documents. The Hypermedia Editor 711 can edit text, visual elements, sound elements, user-input objects, and hotspots for hyperlinks within hypermedia documents. FIG. 3b illustrates how the Hypermedia Editor 711 can be used to lay out user input forms for the various types of subservices that require user input. In the described embodiment, the Hypermedia Editor is used to create and modify documents in the HyperText **Markup** Language (HTML) format. However, any other hypermedia format can also be supported.

Portable Document Editor

The Portable Document Editor 712 is a visual editor for adding hyperlink button hotspots to portable documents. Since portable documents preserve the exact printed appearance of a page, the portable document format is inherently less flexible for on-screen viewing than the format of a hypermedia documents. Thus, only hyperlink buttons can be added to portable documents. For

situations where video, sound, or user input are required, the online service developer should use a hypermedia document instead of a portable document.

Hyperlink Editor

The Hyperlink Editor 714 is a tool that displays and manipulates hyperlinks within an online service. The Hyperlink Editor is described in greater detail in a dedicated subsection, below.

Hypermedia Document Converter

The Hypermedia Document Converter 716 is a Conversion tool that translates documents from various document file formats into a hypermedia document format supported by the Online Designer. For example, the Hypermedia Document Converter can convert various word processor files into HTML files. Once a document is in a hypermedia document format such as HTML, the hypermedia document format can be edited with the Hypermedia Editor 711.

Portable Document Converter

The Portable Document Converter is a conversion tool that translates documents from various document file formats into the portable document format supported by the Online Designer. For example, the Portable Document Converter may translate a Postscript.RTM. file into a portable document format supported by the Online Designer.

Document Harvester

The Document Harvester 718 is a visual tool for specifying which files, directories, and volumes should be indexed for full-text search and retrieval. The Document Harvester displays the file/directory/volume entities in a graphical tree structure such that a developer can specify particular entities for indexing using a cursor control device. In a similar fashion, the developer can specify which specific users or groups of users have the right to access which entities.

Script Editor

The Script Editor 720 is a visual editor for creating and editing the Event Scripts and Function Scripts of an online service. The Script Editor is described in greater detail a dedicated subsection, below. To facilitate quick development of an online system, many sample scripts for many application domains are provided with the Online Designer.

Fee Setter

The Fee Setter 722 is a subtool that specifies how usage fees (if any) should be levied and paid to content providers and users, based on usage of the online service. For example, users can be charged to access information and advertisers can be charged to place advertisements on an online service. The Fee Setter 722 sets fees based upon the usage of the online service. The Fee Setter 722 is the principal subject of this document, and is described in greater detail in a dedicated subsection, below.

Replicator

The Replicator 724 is a subtool that specifies the replication behavior of the content and structure of various subservices. A given subservice may be replicated on multiple servers. Using the Replicator 724, the online service developer can specify: (1) the other servers that participate in replication, (2) which server is the "tie-breaker" when changes on multiple servers conflict, (3) how often a subservice replicates, and (4) whether the subservice content, structure, or both are included in the replication.

Metering Tool

The Metering Tool 726 is a subtool that allows the developer to specify the particular online service usage data that the server should gather. The Metering Tool 726 is described in more detail in its own subsection, below.

Repository Browser

The Repository Browser 728 is a subtool that lists all of the services, subservices, documents, scripts, and other stored resources in the Service Repository, the database associated with the Online Designer. The developer can see the computer disk storage locations for each of these elements, and the amount of disk space occupied. The Repository Browser 728 provides support for moving, copying, and deleting elements within the Service Repository.

Debugger

The Debugger 730 is a subtool that allows the developer to run and debug an online service while the online service is still under development. The Debugger 730 is described in greater detail in its own subsection, below.

Help Editor

The Help Editor 732 is a tool for the developer to author online help for an online service, to be accessed by users. The help information may be context-sensitive such that when the user presses a specific help key or clicks

on a specific help icon, the online service will display appropriate help information for the specific task that the user is attempting to perform. In addition, a general help table of contents and a keyword lookup facility is available for users to search the available help documentation.

In the preferred embodiment, the Online Designer and the related subtools all use similar user interface paradigms, including menus, toolbars, keystroke shortcuts, and mouse techniques such as double-clicking and drag-and-drop. All the design tools also support standard cut, copy, paste, and delete techniques as are well known in the art. For purposes of illustration in what follows, one particular user interface embodiment will be disclosed for demonstrating certain features of the invention. However, it should be understood that the same features can be made accessible using other user interface embodiments. For example, a certain feature can be available both from a pull-down menu and from a toolbar.

Example Online Service

FIG. 8 illustrates a block diagram example of an online service that can be created with the Online Designer. The online service illustrated in FIG. 8 will be used as the basis for a series of examples throughout this document.

The online service structure of FIG. 8 initially shows the user an introductory page for the company. From the introductory page, a user can go to a company personnel directory, a catalog of products made by the company, a list of tips and tricks for using the company's products, a company newsletter, and a listing of corporate information. The example online service illustrated in FIG. 8 can be used as a template for any company that wants to quickly create an online service for its customers.

When the Online Designer is initially invoked, it displays the Service Gallery, which shows all existing online services that are available for modification. FIG. 9 illustrates how the Service Gallery appears to a developer. Each online service is represented by an icon, with the name of the online service displayed beneath. From this screen display, the developer can cut, copy, paste, and delete entire online services.

To "open" an existing online service, the developer double-clicks with the mouse on the icon for the desired online service. This action opens a Service Window, which displays the components of that service. There are four "views" for a Service Window: Connectivity View, Script View, Link View, and Fee View. The initial view for a Service Window is the Connectivity View. To switch between the views, the developer chooses the appropriate pull-down menu item or clicks on the appropriate button on a Service Window toolbar.

The Connectivity View of the Service Window for an online service displays all of the subservices, data sources, and content corpuses that comprise the online service. The Connectivity View also illustrates hyperlinks to other external online services to which this online service connects. For example, FIG. 10 illustrates a Connectivity View for the online service of FIG. 8. As illustrated in FIG. 10, each subservice is displayed with the name of the element subservice beneath.

From the Connectivity View, the developer can cut, copy, paste, and delete entire subservices. The developer can change the data sources (SQL databases, ODBC databases, CD ROM databases, server-based files, data on the local client workstation, etc.) for each subservice; change which content corpuses the service uses within those data sources; and change the hyperlink connections to other online services, including access to all features of the invention's Service-to-Service Protocol. For data sources, the developer can specify and modify the necessary logon information and other parameters necessary for accessing the data.

From the Connectivity View, the developer can double-click on a subservice icon to edit that subservice. Double-clicking on a subservice icon invokes the design tool for that particular type of subservice. For example, double-clicking on the Hyperdocument/Commerce subservice icon for the Company Introduction Page invokes the Hyperdocument Designer tool. FIG. 13 illustrates how the internal structure of the hyperdocument for the Company Introduction Page may appear. As illustrated in FIG. 13, the Hyperdocument Designer tool provides access to the hypermedia documents that comprise the Company Introduction Page Hyperdocument subservice.

Double-clicking on a hypermedia document (shown in FIG. 13) invokes the Hypermedia Editor. FIG. 11 illustrates a Hypermedia Editor view of the Company Introduction Page. The Hypermedia Editor is a What-You-See-Is-What-You-Get (WYSIWYG) editor that displays a hypermedia document as the hypermedia document will appears to an end user. The developer can edit the hypermedia document until the developer is satisfied with its appearance.

The Script View of an online service displays a list of all the scripts in that service. FIG. 12 illustrates a Script View for the online catalog. Each script has a descriptor. The descriptor for an Event Script consists of the name of the event, the name of the visual object to which it applies, the name of the document containing that visual object, and the name of the subservice containing that document. The descriptor for a Function Script is simply the name of the function.

The developer can invoke the Script Editor to view and modify a script by double-clicking on that script's descriptor in the Script View. Alternatively,

the developer can access an Event Script by double clicking on the visual object associated with the script from within the appropriate Designer Subtool. Function Scripts, on the other hand, can only be accessed from the Script View.

When the developer chooses the Link View of an online service, the Hyperlink Editor is invoked to view and modify the hyperlinks between the subservices of the online service. For hyperlinks between individual documents and files, see the Link View of the Hyperdocument Designer, below. For hyperlinks between whole services, see the Connectivity View of the Online Designer, above.

The Fee View of an online service provides access to the Fee Setter subtool. When invoked from the Connectivity View, the Fee Setter subtool allows the developer to specify the cost (if any) of accessing the service as a whole. To specify fees for individual documents or parts of a service, the developer invokes the Fee Setter subtool from the individual Designer Subtools such as the Hyperdocument Designer Subtool.

At any time while running the Online Designer, the developer may access the Replicator tool to specify replication behavior among services and subservices. The Metering Tool can be accessed to specify service and subservice metering characteristics, and the Repository Browser can be used to view and manipulate the contents of the Service Repository. In addition, the developer may invoke the Debugger to run and debug an online service or a single subservice.

Key Designer Subtools

Each of the Designer Subtools is used to develop a particular type of subservice within an online service. The most important and original Designer Subtools are: (1) the Hyperdocument Designer, for subservices that consist of linked hypermedia and portable documents; and (2) the Lookup Designer, for Directory Lookup, Classified Advertisement, and Reference subservices. This section describes these two Designer subtools in detail.

The Hyperdocument Designer Subtool

The Hyperdocument Designer 412 subtool is used to design Hyperdocument/Commerce subservices. Specifically, any subservice that displays hyperdocuments and supports hyperlinks between hyperdocuments is designed using the Hyperdocument Designer subtool. A typical online service will require these features to some degree, so most online services include at least one Hyperdocument/Commerce subservice.

When the developer double-clicks on a Hyperdocument/Commerce subservice icon from the Connectivity View of the Service Window for an online service, the Hyperdocument Designer is automatically invoked. Hyperdocument Designer

supports four different views: Document View, Script View, Link View, and Fee View. FIG. 14 illustrates a block diagram of the different views supported Hyperdocument Designer. To switch between views, the developer chooses the appropriate menu item or clicks on the appropriate button on the Hyperdocument Designer toolbar.

Initially, Hyperdocument Designer displays the Document View, which shows one icon for each of the hyperdocuments that comprise the subservice. An example of a hyperdocument being viewed with the document view is illustrated in FIG. 13. Beneath each icon is the name of the document. The document icons are visually shown in the arrangement laid out by the developer. The developer may rearrange the document icons in the Document View using drag-and-drop mouse techniques, and the icon arrangement is preserved between sessions with Online Designer. The icon arrangement in the Document View is for developer convenience only, and has no bearing on the behavior of the hyperlinks that define the order in which the user sees documents. Double-clicking on a hypermedia document icon or portable document icon invokes the Hypermedia Editor or Portable Document Editor, respectively, to view and modify that document.

From the Document View, the developer may invoke the Hypermedia Document Converter or Portable Document Converter from the Hyperdocument Designer menus or toolbar. These two Converter tools translate preexisting documents from various file formats into the Online Designer's standard hypermedia document format or portable document format. After translating a document, the developer assigns a name and icon to the new hyperdocument, and repositions the new icon within the Document View as desired. The developer may then edit the hyperdocument to add visual objects using the Hypermedia Editor. The developer can connect the new hyperdocument using the hyperlink editor to edit hyperlinks that integrate the document into the subservice.

The Script View, Link View, and Fee View of the Hyperdocument Designer are analogous to the same views in the Online Designer tool itself. The Script View provides access to the Event Scripts and Function Scripts that pertain to the particular Hyperdocument/Commerce subservice. When the developer chooses the Hyperdocument Designer's Link View, the Hyperlink Editor is invoked to view and modify the hyperlinks between all documents and files associated with the subservice. The Fee View of the Hyperdocument Designer invokes the Fee Setter subtool to specify fees for individual documents and files in the subservice.

In effect, the Hyperdocument Designer is a Designer Subtool that provides organized access to the Utility Subtools that are most often used in designing a hypermedia and/or commerce subservice. For example, suppose that an existing mail-order catalog shopping company wishes to use the invention to design and deploy and online service that is the electronic equivalent of an existing

mail-order catalog service. The developer could invoke the Hyperdocument Designer to create a new Hyperdocument/Commerce subservice.

The developer could use the Portable Document Converter to convert an electronic version of the company's catalog into a portable document. Using the Portable Document Editor, the developer could add hyperlink buttons to the portable document, which lead the user to the electronic order form from the catalog. From the Document View, the developer could invoke the Hypermedia Editor to create an electronic order form with a button to submit the order. FIG. 15 illustrates a hyperdocument that displays information about a product. The hyperdocument of FIG. 15 includes a "Place Order" button 1530 that moves the user to a purchase order screen.

FIG. 16 illustrates a purchase order screen that could be connected to the "Place Order" button with a hyperlink. The purchase order screen allows an end user to enter information to order the product. After the user has entered the necessary information, the user selects the "Purchase" button to buy the product. An event script processes the information and orders the product. To edit the event script, a developer double-clicks on the "Purchase" button 1630 from the Hypermedia Editor. The developer edits the script associated with the "Purchase" button 1630 script such that the script gathers the information from the form and submits the order as a transaction against the back-end order/inventory database system. When the design is complete, the deployed service allows users to view the catalog online, and place orders in real-time, without human intervention.

The Lookup Designer Subtool

The Lookup Designer tool designs Directory Lookup, Classified Advertisement, and Reference subservices. In each of these types of subservices, the user can search through a database of entries, and in some cases, the user can submit new entries. The differences between the subservice types include the kinds of visual objects found in the entries, the browsing and searching techniques supported, and whether or not the user can submit new entries for public viewing.

When the developer double-clicks on one of the three types of lookup subservices in the Connectivity View of the Service Window for an online service, the Lookup Designer is automatically invoked. As illustrated in FIG. 17, the Lookup Designer supports six views: Form View, Category View, Options View, Script View, Link View, and Fee View. As with the other Designer Subtools, the developer uses menus or the Lookup Designer toolbar to switch between the different views.

The initial view for Lookup Designer is the Form View. From the Form View,

the developer uses the Hypermedia Editor to design the Submit Form, View Form, and Query Form for the subservice.

The Submit Form allows a user to submit a new entry to be listed on the subservice. FIG. 18 illustrates an example screen display for a submit form for the Corporate Personnel Directory subservice of FIG. 8.

The View Form is the template that displays the contents of an entry to the user. FIG. 19 illustrates an example screen display for a view form for the Corporate Personnel Directory subservice of FIG. 8.

The Query Form allows the user to search for entries based on various criteria. FIG. 20 illustrates an example screen display for a query form for the Corporate Personnel Directory subservice of FIG. 8.

There are certain standard input fields that the various Lookup Designer forms may provide. A form need not use all of the standard fields. However, for the standard fields that are used, the form should also use the standard internal names for those fields so that the fields will be properly recognized and handled by Molisa. If a given form does not yet exist for a subservice, Online Designer provides a default form containing all standard supported fields, which the developer can modify.

The standard Lookup Designer form fields are as follows:

Name For Directory Lookup subservices, this is the person or entity name for the entry. For Reference subservices, this is the name of the item for which reference information is being provided. For Classified Advertisement subservices, this is the name of the person submitting the entry.

Address, Phone, Fax, **E-mail** For Directory Lookup subservices, these fields pertain to the person or entity listed in the entry. For Classified Advertisement subservices, they pertain to the person submitting the entry. These fields are typically not used for Reference subservices.

Categories The categories under which the entry should be listed. Typically, this is a pairing of two visual objects: a drop-down list box showing all of the categories recognized by the subservice, and a text entry field that displays the list of categories that the user has chosen so far for this entry. Note that the user must choose from among the categories specified by the developer in the Category View of the Lookup Designer when the subservice was designed. (The developer can of course change the list of categories at a later time using Online Designer.) This field is typically used in Classified Advertisement and "yellow pages" Directory Lookup subservices. The categories are used for browsing purposes, in that all entries that belong

to a given category are shown together under that category name. In Classified Advertisement subservices, the categories can be used as part of the search criteria on a Query Form.

Keywords Similar to categories, but users can type in any keywords they deem appropriate, rather than being constrained to choose from a fixed list. Entries sharing a common keyword are not listed together in the subservice browser, but users can search the keyword fields of entries using the Query Form.

Slogan Advertising slogan. This field is typically only used in "yellow pages" style Directory Lookup subservices.

Description The descriptive text for the entry. For Classified Advertisement subservices, this is the text of the advertisement itself. For Reference subservices, this is the information about the named entry. For "yellow pages" style Directory Lookup subservices, this is the description of company products and services, hours of operation, etc. This field is typically not used for "white pages" style Directory Lookup subservices.

Image A graphic image to be displayed with the entry. In the Submit Form, this is a text field for the user to specify a directory path to the file containing the image. In the View Form, this is a picture field that displays the image itself. This field is typically used in "yellow pages" Directory Lookup subservices for a company logo or related graphic, or optionally in a Reference subservice for a picture of the named item.

Removal Date Date after which an entry should be automatically removed from the subservice. This is typically used for Classified Advertisement subservices, but it can also be used in Directory Lookup subservices to reduce costs for the submitter. A Classified Advertisement subservice can also have a global automatic limit on the number of days that an entry is listed (see the Options View, below).

Service Link Used for entries that provide a hyperlink icon leading to another online service. For example, a "yellow pages" entry can provide a link to an online service provided by the company listed in the entry. On the Submit Form, this is a text field for the user to provide the **URL** of the other online service. On the View Form, this is displayed as the hyperlink icon itself.

In addition to the standard Lookup Designer form fields, the developer may include other input fields that have specific meaning to the subservice being developed. Such fields make it easier to query the database of entries. For example, a Classified Advertisement subservice devoted to the purchase and sale

of pre-owned automobiles can include form fields for the year, make, and model of the car. The end user can type specific information into those fields on the Query Form to find a matching entry, instead of performing a less-precise full-text search on the Description field of the entry.

The developer should associate any required scripts with the visual objects on a form, to specify the behavior of those visual objects. For example, on a Query Form, the developer should provide a script for the Search button on the form that specifies how to convert user input in the various fields of the form into a query against the subservice database of entries. To invoke the Script Editor to create a script for a visual object, the developer double-clicks on that visual object while viewing the associated form in the Hypermedia Editor.

The Category View of the Lookup Designer simply displays a list of the category names supported by the subservice. The developer can add, delete, and modify the categories from the category view.

In the Options View of the Lookup Designer, the developer specifies certain options about the behavior of the subservice. These include:

Updatable A checkbox that indicates whether users can submit new entries to the subservice for other users to view. If this box is not checked (e.g., for a Reference subservice), any entries submitted by a user can later be viewed by that user, but no one else.

Moderated A checkbox that indicates whether this subservice is moderated. If this box is checked, any newly submitted entries are not directly posted to the subservice. Instead, they are transparently transmitted by electronic mail to the moderator for the subservice. The moderator reviews the entry, and if it is deemed appropriate, the moderator posts the entry on behalf of the original submitter. This option is ignored if the Updatable checkbox is not checked.

Categorized A checkbox that indicates whether this subservice supports browsing by category. If this box is checked, all entries in a common category are grouped together under that category name for browsing by users. If a single entry is in more than one category, it appears under each of those categories. Typically, this box is checked for "yellow pages" Directory Lookup subservices and Classified Advertisement subservices. Whether or not this box is checked, the user may still perform standard queries against the database of entries using the Query Form for the subservice.

Sorting A drop-down list box that indicates how entries are sorted within a category for user browsing. Entries may be sorted in forward or reverse order based on the contents of any of the entry fields, and secondary and tertiary

sort keys are supported with additional drop-down list boxes in the tool. In addition, the entry's date of posting is available as a sort key, even if that date is not displayed as part the entry itself. Typically, a "yellow pages" Directory Lookup subservice will sort in forward order based on the contents of the Name field, and a Classified Advertisement subservice will typically sort in reverse order of posting date.

Expiration The number of days that each entry remains listed on the subservice before it is automatically removed. This option may also be left blank, in which case there is no automatic expiration date for entries. If an entry has an individually specified Removal Date that occurs before automatic expiration, the entry's Removal Date is honored. Otherwise, the automatic expiration date is used.

The Script View and the Link View in the Lookup Designer are analogous to the Script View and the Link View in the Online Designer tool itself and in the Hyperdocument Designer.

The Fee View of the Lookup Designer is an optional feature that invokes the Fee Setter subtool, allowing the developer to specify the formula for computing the cost of viewing an entry (if any), and submitting an entry (if any). The Fee Setter subtool is described in greater detail in a separate section below.

Utility Subtools

The Utility Subtools provide capabilities that are useful in the design of multiple types of subservices. These subtools are accessed from the Designer Subtools, and from the Online Designer itself. The most significant and original Utility Subtools are: (1) the Hyperlink Editor, for manipulating hyperlinks within an online service; (2) the Script Editor, for editing the various scripts that control the behavior of an online service; (3) the Fee Setter, which allows the developer to specify any fees that should be charged to users or advertisers; (4) the Metering Tool, which provides instructions to the online service server regarding the usage statistics that should be tracked; and (5) the Debugger, which provides interactive running and debugging capabilities for an online service. This section provides the details on the five Utility Subtools.

The Hyperlink Editor Subtool

The Hyperlink Editor Subtool is a Utility Subtool that is used to display and manipulate hyperlinks within an online service. The hyperlinks within an online service can be viewed and modified at various levels of abstraction. For example the Hyperlink Editor Subtool can display and manipulate the links between different subservices within an online service, the links between

different documents within a subservice, the links within a single document, or the individual attributes of the links themselves.

With the Hyperlink Editor, the developer can assign attributes to hyperlinks. Some important examples of hyperlink attributes include: (1) whether the hyperlink leads to the same document/file or to a different document/file; (2) whether the hyperlink leads to the same online service or a different online service; (3) whether the hyperlink leads to a service that the developer controls or doesn't; (4) the size of the document/file a link points to; (5) whether the link leads to a free service or one that has additional charges; and (6) the semantics of the hyperlink. The latter attribute is a semantics tag taken from a known list of possibilities, which includes simple linking, making a purchase, returning to the home page of the service, initiating a search, linking to a form that requests shipping address information, etc. The semantics attribute of hyperlinks provides some additional structure to online services, and encourages a degree of standardization in hyperlink usage.

The Hyperlink Editor supports both a Graphical View and a List View of hyperlinks. The Graphical View displays the hyperlinks as a directed graph, with the source and target of a hyperlink represented by visual icons, and displays the hyperlink itself as a directed arc connecting them. An arc's particular appearance (color, width, arrow design, and other visual cues) depends on the various attributes (above) associated with that hyperlink. The List View displays a list of the hyperlinks, showing the names of the linked entities and visual cues indicating the attributes associated with each hyperlink. The developer can modify the hyperlinks and attributes from either view.

A search facility within the Hyperlink Editor allows the developer to search through the online service for hyperlinks that satisfy a list of criteria. The search criteria are expressed as a set of attribute values associated with the hyperlinks, which the developer types into a search query form.

The Script Editor Subtool and the Online Script Language

The Script Editor is a Utility Subtool for editing the Event Scripts and Function Scripts of an online service. The script editor is accessed from the Script Views of the various Designer Subtools and the Online Designer itself. The developer can also invoke the Script Editor to edit the Event Scripts associated with a visual object by double-clicking on the visual object itself from within the appropriate Designer Subtool.

For example, FIGS. 21a and 21b illustrate screen displays for the Script Editor editing an event script for the for the MouseDown event on the

"Purchase.sub.-- Button" of the hyperdocument illustrated in FIG. 16. The Purchase event script of FIGS. 21a and 21b checks the inventory, charges the customer, and updates the inventory if a sale is completed.

Event Scripts and Function Scripts conform to the invention's Script Language, a procedural programming language similar to the language BASIC (Beginner's All-Purpose Symbolic Instruction Code). The Online Designer Script Language includes variable declarations, numeric and string operations, conditional statements ("if . . . then . . . else"), control statements for looping ("for" and "while"), and functions (subroutines) with parameter passing. The Script Language includes a number of programming constructs and built-in functions. The programming constructs and built-in functions are collectively referred to as the Script Language "primitives". The primitives included in the Script Language have been chosen and optimized for implementing common features supported by online services.

Using the Script Editor, a developer can directly type and edit an event or function script. In addition, the script editor provides a menu-driven facility to paste script statement and function invocation archetypes into a script, which the developer can then modify appropriately. For example, the developer can use the menus to insert a "for" statement archetype that has placeholders for the conditional expression and statement body. Similarly, archetypes for all of the built-in functions and programming constructs can be inserted, with placeholders for the various function arguments.

Many of the key features of the present invention are accessed primarily through the primitives of the Script Language. In addition to normal programming language primitives for arithmetic, file input/output, etc., specific primitives are included to support for online services. The following sets of script primitives exist to support online services.

Program control primitives

A set of primitives are provided for transferring program control to another document, subservice, or service. This is the dynamic form of a hyperlink. By using these primitives in a script, the developer can choose the destination of a hyperlink at "run-time," in response to previous input from the user, or depending on the context in which the particular hyperdocument is displayed.

Telecommunication primitives

A set of primitives exist for performing various telecommunication tasks such as downloading files to a client system. For example, in electronic publishing, the user can click on a button to download the electronic version of a magazine. The Event Script associated with the "Mouse Down" event on that

button would invoke the primitive to download the document. As another example, a script can invoke the primitive to download software for viewing JPEG **compressed** images if it does not find that external viewer already resident on the client workstation.

Text search and retrieval primitives

Primitives are provided to specify the various types of text search criteria: natural language, Boolean, and conceptual. Other primitives are provided to initiate the search, using the previously specified criteria, against specified data sources. For example, to perform a search based on the contents of a query form, a script should construct an appropriate Boolean text query from the keywords typed into the user input fields on the form, and submit that query using the Boolean criteria language primitives. Then, the script should invoke the built-in function that initiates the search, passing an argument to that function that specifies the ID of the target database for the search.

External database access primitives

Direct access to external databases and real-time data is provided using specific script primitives. The external database primitives provide the most common and standardized constructs supported by Structured Query Language (SQL), to access relational database systems. In addition, the Script Language includes a general SQL primitive that can accept any sequence of native SQL statements, either as an argument to the function or contained in a specified file, and pass those SQL statements directly to the relevant database system as illustrated in FIG. 22. Any results from the database query are returned to the subservice that made the query. For non-SQL databases (and certain SQL databases), there are primitives for Open Database Connectivity (ODBC).

External communication primitives

A set of primitives exist that allow an online service to communicate with other programs and users. For example, the primitives are provided with allow a script to send an electronic mail message, a facsimile transmission, a **voice mail** message (using text-to-voice techniques), or a message to an electronic **wireless** pager.

Application program control primitives

The Script Language provides various primitives for launching other application programs, sending data to those programs, and receiving data from those programs. The names and semantics of the external program control primitives differ according to the server platform that will run the online

service.

For example, under a Microsoft Windows NT Server Version of the Online Designer, there are Script Language primitives for Dynamic Data Exchange (DDE), launching another application with optional keystroke stuffing, and batch invocation. To support an online service acting as a DDE client, there are primitives for standard DDE Connect, Disconnect, Execute, Poke, Request, Advise, and Unadvise actions. In addition, the developer may write Event Scripts that trigger on DDE Advise events from other programs.

For launching and keystroke stuffing, the Script Language provides primitives that launch a named software application, optionally wait for the application to finish or let it run concurrently, and optionally "stuff" specified keystrokes into the launched application with suitable pauses at specified points in the keystroke stream. The batch primitives simply launch a batch file or batch-style application with a given command line, and the Script Language file I/O primitives can then be used to read and parse the output of the batch process (if any). For other server platforms (OS/2, UNIX, etc.), other application control primitives are provided that are appropriate on those platforms. For example, terminal emulation primitives are one mechanism provided under UNIX.

External control primitives

Script Language primitives are provided that send commands to, and receive data from, electronically controlled equipment such as heating systems, ventilation systems, air-conditioning systems, security systems, and lighting.

Access control primitives

Using the access control primitives of the Script Language, a script can request a password from the user, encrypt and decrypt files to be downloaded or sent to another service, and dynamically determine whether a given user should be given access (read only, write only, read/write, or none) to a particular document or part of a service. These run-time language primitives augment the static access control mechanisms that the developer can specify at design time using the invention.

Service-to-service communication primitives

The Script Language provides service-to-service communication primitives that allow one online service to: (1) act on behalf of the user to query or update another online service; (2) automatically update another online service without user initiation; (3) appear to be seamlessly part of another online service; (4) keep a record of how many times users traverse to another online

service; (5) pass along automatic user registration data to another online service; (6) automatically register a new online service with a service-of-services or "yellow pages" service; (7) check whether another server is running a particular online service or type of service; and (8) exchange usage and metering information, for aggregation and later analysis. Each of these primitives opens a virtual connection to the target service, using the service-to-service protocol.

User communication primitives

User communication primitives exist that allow users to engage in real-time cooperative activity. These user communication Script Language primitives provide a Named Pipes style communications interface between two or more users, or between users and a representative of the online service provider. For example, such primitives can support a multi-person game between users, or a user entering an online query and receiving a real-time response from a representative of the service provider. The primitives to establish a connection include the ability to specify a specific user with whom to communicate, or a "broadcast" facility to find any current user on a given server who wishes to establish a given class of cooperative connection.

Image capture primitives

Primitives are provided that: (1) command the server software to accept a facsimile that is sent to the server's fax modem from a user's fax modem or fax machine having a given identification, (2) command the server software to accept an image transmitted by electronic mail to the server, or (3) command the client software to accept a scanned image (using the TWAIN scanning standard) and transmit that image to the server. Once captured, other Script Language primitives allow the image data to be incorporated into other documents. For example, a logo for a "yellow pages" listing or a photograph for an online classified advertisement.

The Script Editor works cooperatively with the Debugger, to allow single-stepping through scripts, displaying script variables, etc. In debug mode, the Script Editor allows certain limited changes to a script. More major script changes require that the developer stop the simulation first.

The Fee Setter Subtool

Fee Setter Introduction

The Fee Setter subtool allows the developer of an online service to specify the fees that will be levied on or paid to users, as users use the service and access the information it contains. The Fee Setter subtool of the Online

Designer can also be used to define fees levied on or paid to information content providers. The online service framework automatically levies and pays the fees according to the Fee Setter instructions, on behalf of the organization that operates the online service.

The actual transfer of monetary funds specified by the Fee Setter can be effected on an immediate or periodic basis using mechanisms external to the Fee Setter itself. For example, the Fee Setter can use credit card charges or electronic funds transfer to charge users of an online service. Similarly, the Fee Setter can use electronic funds transfer or traditional paper-based billing and payment mechanisms to bill content providers, or other similar means. The Fee Setter specifies the fees to be levied and the payments to make, and the external mechanisms arrange the funds transfer to actually cover those fees and payments.

The Fee Setter is used for all of the various chargeable entities in an online service. As such, the Fee Setter is accessible from the Fee Views of the Online Designer itself and from the Fee Views of each of the Designer Subtools. For example, the Fee Setter can be accessed from the Fee View of the Hyperdocument Designer to charge fees for documents and for following hyperlinks to other subservices. Similarly, the Fee Setter can be accessed from the Lookup Designer to charge fees to users who view/download entries and those who submit new entries.

Many fees are simply constant monetary amounts; e.g., a $2.00 fee to download a particular document. Other fees are more complex. The fees can depend on the size of a document, the time of day, the load on the server, the identity of the user, the number of previous documents downloaded or submitted by this user, etc., depending on the information provider's policies and intentions. For classified advertisement submissions, the fee for submitting the advertisement can include the size of any graphic image in the advertisement, the number of categories under which the advertisement is listed, and the number of days that it is listed.

The Fee Computation in a Fee Specifier supports simple and complex fee structures. The formula itself is specified using a subset of the online service Script Language. When writing the Fee Formula, the developer writes a sequence of script statements, possibly including variable declarations, such that the appropriate fee is assigned to the reserved global variable "Fee@" some point before the end of the script.

Example Fees

To illustrate some of the types of fee structures that can be created using the Fee Setter, several examples of fees that can be defined with the Fee

Setter are listed below. It should be understood that these are examples only, and that many other types of fee structures can be created using the Fee Setter of the present invention:

Levying Fees on Users

Levying a fixed fee on users whenever certain textual or graphic information is viewed or downloaded from the online service.

Levying a variable fee on a user for accessing information, depending on the amount of information that particular user has accessed in the past. Thus, a quantity discount can be offered to users that frequently access a particular online service.

Levying a variable fee on users for accessing information, wherein the fee charged depends on the time of day that the information is accessed or on the current load on the online service server. Thus, the amount of the fee would discourage access during peak periods by assigning a premium during peak hours.

Levying variable fees on users depending on the size of the information accessed, or on the amount of time required to access, view, or download the information.

Levying different fees on different classes of users. For example, users that have paid for an annual membership will receive a discount.

Levying a fee on users for simply connecting to a given online service. For example, an online service that provides investment advice could charge for access.

Levying a fee on users who access certain parts of an otherwise free online service. For example, in an online service provided by a free newspaper publisher, a fee could be charged for users who wish to access the full-text search capabilities on back issues.

In a classified advertising online service, levying a variable fee on users who electronically submit new listings to the service, depending on the size of the listing.

Paying Fees to Users

Paying a fixed fee to a user in exchange for that user filling out a market survey questionnaire.

Paying a fixed monetary prize to a user as winnings from a contest run by

the online service operator.

Paying a variable fee to a user as proceeds from (legal) gambling conducted on the online service. Thus, users could engage in gambling from home.

Levying Fees on Content Providers

Levying a fixed fee on a content provider whenever a user views or downloads that provider's textual or graphic information from the online service. For example, when a user views or downloads a content provider's advertising brochure or investment prospectus, a small fee would be levied on the content provider.

Levying a variable fee on a content provider when a user accesses the provider's information, depending on the amount of information that all users have accessed from that provider in the past. Thus, an advertising quantity discount to the content provider.

Levying variable fees on content providers depending on the amount or size of information carried on the online service, and on how many days that information is carried on the service.

Levying different fees on different classes of content providers. For example, an online service provider could give a discount to non-profit organizations that advertise on the online service.

Levying a fee on another online service provider whenever a user clicks on a hyperlink from the current online service that leads to the content provider's own online service. This would in effect be a referral fee.

In a "yellow pages" style online service, levying a variable fee on a content provider depending on the number of categories under which the provider's listing (and advertisement) is carried. Thus the easier it is to find the content provider's advertisement, the more the online service provider would charge.

Paying Fees to Content Providers

Paying a fixed fee to a content provider whenever a user views or downloads a particular document or program posted by that content provider. Thus content providers can supply informative reports, software programs, images, sounds, etc. that would be available to users of the online service. When a user requests the content provider's material, the users would be a charged for the material and the fees charged to the user would be divided between the content provider and the online service operator.

Paying a variable fee to a content provider depending on the size of the provider's textual or graphic information that is downloaded by all users.

Paying a variable fee to a content provider when users perform full-text searches across the provider's database of documents. The fee paid to the content provider depends on how much time was spent performing searches (even if no documents were ultimately viewed or downloaded).

Paying a variable fee to a content provider of (say) stock photo images when an end-user downloads an image, where the fee depends on the total number of images downloaded by all end-users in the past; in effect, a quantity discount to the online service operator on paying for content.

Fee Specifiers

The Fee Setter allows the developer to use the mouse, toolbar, and menus to create, modify, and delete individual Fee Specifiers. A Fee Specifier is a tuple that consists of four different parts: (1) An action that triggers the fee. This can be a traverse of a hyperlink, the downloading of a document, the uploading of an advertisement, etc.; (2) The argument values (if any) that are required by the specified action; (3) The entity to whom the fee should be charged or to whom the payment should be made. This can be a user of the online system or a content provider; (4) A Fee Computation that specifies exactly how a fee or payment is computed.

The Fee Setter displays the Fee Specifiers in a list. For example, FIG. 23 illustrates a list of Fee Specifiers that can be used to assign fees in one particular online service. Each Fee Specifier describes one particular type of fee for using the online service. The Fee Computation is not displayed directly in the Fee Setter list. Instead, in each Fee Specifier, an on-screen button is displayed that can be clicked by the developer to access the Script Editor to view and edit that Fee Computation. FIG. 24 illustrates a Computation Script Editor view of a Fee Specifier.

The detailed descriptions of each element in the Fee Specifier tuple are given below:

Action: This is the type of action that triggers the fee to be charged or the payment to be made. The allowable Action values are:

Access The action of a user accessing (viewing, downloading, "running") an object. The supported objects are: document, image, video clip, sound clip, and script.

Submit The action of a user submitting (uploading) an object. The supported objects are: document, image, video clip, and sound clip.

Traverse The action of a user clicking on (traversing) a hyperlink in a particular document.

Connect The action of connecting to the online service.

Daily Indicates that the fee or payment is recomputed and reassessed once each day.

Weekly Indicates that the fee or payment is recomputed and reassessed once each week.

Monthly Indicates that the fee or payment is recomputed and reassessed once each month.

Annually Indicates that the fee or payment is recomputed and reassessed once each year.

Argument: This is the argument value (if any) required by the action element of the Fee Specifier. The Argument values required by each of the various types of Action are as follows:

Access The file system path of the affected object. For example, "/pub/www/clothing/order.html."

Submit The file system path of the document form that was used to submit the object. For example, "/pub/www/classifieds/newlisting.html."

Traverse The hyperlink being traversed. It is specified as the path of the document containing the hyperlink, followed by three slashes ("///"), followed by the **URL** of the hyperlink itself as contained in that document. For example, "/pub/www/yellow/acme.html///http://www.ac me.com."

Connect Requires no argument. The argument field should be left blank.

Daily Requires no argument. The argument field should be left blank.

Weekly Requires no argument. The argument field should be left blank.

Monthly Requires no argument. The argument field should be left blank.

Annually Requires no argument. The argument field should be left blank.

Entity: This is the entity to whom a fee should be levied or paid from the online service operator. The allowable Entity values are:

Provider The fee should be levied or a payment should be made to the content provider.

User The fee should be levied or a payment should be made to the user.

Note that if the action element of the Fee Specifier is "Daily" the entity element is "Provider", the Fee Specifier is recomputed and reassessed for every content provider in the online service each day. Similarly, if the action is "Daily" and the entity is "User", then the Fee Specifier is recomputed for every user of the online service each day. The same principle holds for fees that have the action triggers of "Weekly", "Monthly", and "Annually."

Fee Computation. This is a script, written in the Computation Language, that specifies how the fee to be levied or paid is to be computed. Details of the Fee Computation are provided in a separate section below.

Example Fee Specifiers

To best illustrate the use of Fee Specifiers, two examples of Fee Specifiers are listed below

_____ Example Fee Specifier #1
_____ Access,
/pub/www/research/crop-forecast-94.html, User, Fee@ = 5.00 >
_____

This first Fee Specifier indicates that a fee should be charged whenever a user accesses (views or downloads) the document identified by the path "/pub/www/research/crop-forecast-94.html," since the action is "Access" and the argument is the "/pub/www/research/crop-forecast-94.html" path name. When a users performs such an access, the online service should levy a fee of $5.00 on the user.

_____ Example Fee Specifier #2
_____ Daily, Provider, Fee@ = 0.0 For i% =
1 To ProviderFileCount%(Provider%) Fee@ = Fee@ + (1E-6 *
FileLen(ProviderFilePath$(Provider%, i%))) Next i% >
_____

This second example Fee Specifier indicates that each day, a particular fee should be charged to each content provider. The daily fee is calculated by charging $0.000001 for each byte of file data that is owned by that content

provider on the online service. Stated more simply, each content provider is charged approximately $1.00 for each megabyte of data stored on the online service per day. Note again that the list of Fee Specifiers shown in the Fee Setter do not actually contain the entire Fee Computation above. Instead, an on-screen button in the Fee Specifier tuple can be clicked to launch the Script Editor, allowing the developer to view and modify the Fee Computation.

The online service framework automatically executes the appropriate Fee Specifiers whenever their associated actions occur. Thus, when the user accesses a document, the "Access" Fee Specifiers (if any) whose argument element is the path to that document will be executed, resulting in a fee being levied or paid for each such Fee Specifier. Once a day the "Daily" Fee Specifiers are executed, and so on.

Note that, in many cases, a Fee Specifier that levies a fee on a user for accessing information will be accompanied by another Fee Specifier that pays part of that fee back to the content provider. Conversely, a Fee Specifier that pays a fee to the user (e.g., for filling out a market survey questionnaire) will often be accompanied by a Fee Specifier that levies a comparable fee on the content provider.

Fee Computation

The Fee Computation in a Fee Specifier supports simple and complex fee structures. Each Fee Computation is expressed using the Computation Language, which is a subset of the online system development tool's full Script Language. When specifying the Fee Computation, the developer writes a sequence of script statements such that when the script is executed by the server, the appropriate fee is assigned to the predefined global variable "Fee@" at some point before the end of the script. If the final value of Fee@ is positive, then a fee is levied on the entity by the online service operator; if it is negative, the fee is paid to the entity by the service operator.

Computation Language Basics

The Computation Language is a subset of the invention's Script Language, which is itself similar to the computer programming language BASIC (Beginner's All-purpose Symbolic Instruction Code). A very brief overview of the Computation Language is provided below. The reader is referred to any comprehensive reference on the BASIC programming language for a description of the detailed semantics of these programming constructs.

_____ Action Statement Syntax
_____ Expression evaluation Operators (+, -, *, /, etc.) and function calls, in the usual fashion Variable assignment

<variable> = <expression> Conditional execution If <condition> Then <statements> [Else <statements>] Endif Repeated execution Do While <condition> <statements> Loop Iterative execution For <variable> = <expression> To <expression> <statements> Next <variable>

In the Computation Language of the present invention, explicit variable declarations are not used or required. Instead, the suffix character used on a variable name determines the data type of the variable:

Suffix	Data Type	Range	Example Constants
%	Integer	+/-2,147,483,647	3019, -12
!	Floating pnt	+/-4.94 .times. 10.sup.-324 to +/-1.79 .times. 10.sup.308	-2.25, 2.879E-35
@	Currency	+/-922337203685477.5807	562.91, -0.1822
$	String	0 to 65,500 characters	"Hello",""
#	Date/Time	01-Jan-0000 0:00 to 31-Dec-9999 23:59	#02-Apr-94 1:45 pm#

The Currency data type (variables with the @ suffix) supports up to 4 digits to the right of the decimal point. It maintains exact decimal accuracy, making it especially suitable for monetary calculations. In the Date/Time data type, the base unit is days such that adding or subtracting an integer adds or subtracts days; adding or subtracting a fraction adds or subtracts time as a fraction of a day. For example, adding 10 adds 10 days, while subtracting 1/24 subtracts one hour.

Predefined Global Variables

There are 5 predefined global variables available to a script that comprises a Fee Computation. The 5 predefined global variables are defined below:

Fee@ When the Computation Language script that defines the Fee Computation is complete, the final value of the Fee@ predefined global variable is the fee that is levied on the entity (if the value is positive) or paid to the entity (if the value is negative).

Arg$ The value of the argument element of the Fee Specifier. The Arg$ is provided as a notational convenience.

Provider% The provider identifier number of the content provider associated with the action that triggered the Fee Specifier. This predefined global

variable is available when the entity element of the Fee Specifier is "Provider". If the action is "Access", the value of Provider% is the identifier number of the content provider that owns the information that was accessed. If the action is "Daily", "Weekly", "Monthly", or "Annually" (and the Fee Specifier is "Provider"), the Fee Specifier is evaluated once for each content provider of the online service. In this case, the Provider% value is the provider identifier number of the current content provider being referenced in this iteration of the Fee Specifier computation.

User% The user identifier number of the user associated with the action that triggered the Fee Specifier. This predefined global variable is available when the entity element of the Fee Specifier is "User". If the action is "Access" or "Submit", the value of User% is the ID number of the user that accessed or submitted the information. If the action is "Daily", "Weekly", "Monthly", or "Annually" (and the Fee Specifier is "User"), the Fee Specifier is evaluated once for each user of the online service. In this case, the User% value is the user identifier number of the current user being referenced in this iteration of the Fee Specifier computation.

Access Time# The amount of elapsed time that was required to access the current object. This predefined global variable is valid only in "Access" or "Submit" Fee Specifiers.

Available Built-In Functions

All of the primitives of the Script Language are available in the Computation Language. These primitives include built-in functions for general computing purposes (e.g., "Now()" to obtain the current date/time, "FileLen(<path>)" to determine the length of a file, etc.), as well as built-in functions that are specific to online services. The online service primitives that are of particular interest for creating Fee Computations are detailed below:

ProviderFileCount%(<provider.sub.-- num>)

Returns the total number of files, carried on the online service, belonging to the content provider whose provider identifier is <provider.sub.-- num>.

ProviderFilePath$(<provider.sub.-- num>, <index>)

Returns the path of the file at index <index> in the list of files associated with the content provider whose provider identifier is <provider.sub.-- num>. The allowable range of <index> is 1 through ProviderFileCount%(<provider.sub.-- num>), inclusive.

ProviderTotalAccessCount%(<provider.sub.-- num>)

Returns the total number of files, belonging to the content provider whose provider identifier is <provider.sub.-- num>, that have ever been accessed by any users on this online service. The ProviderTotalAccessCount% function is useful for computing quantity discounts.

ProviderTotalAccessSize%(<provider.sub.-- num>)

Returns the total size of the files, belonging to the content provider whose provider identifier is <provider.sub.-- num>, that have ever been accessed by any users on this online service. The ProviderTotalAccessSize% function is useful for computing quantity discounts.

ProviderTotalContentCount%(<provider.sub.-- num>)

Returns the total number of files on this online service belonging to the content provider whose provider identifier is <provider.sub.-- num>.

ProviderTotalContentSize%(<provider.sub.-- num>)

Returns the total size of all files on this online service belonging to the content provider whose provider identifier is <provider.sub.-- num>.

ProviderAttrSet(<provider.sub.-- num>, <attr.sub.-- name>, <value>)

Associates, with the content whose provider identifier is <provider.sub.-- num>, an attributed name <attr.sub.-- name> having value <value>. If that attribute already exists for that provider, replaces the value of the attribute with this new value. One example of using attributes on content providers might be to record in a "Non-profit" attribute the value "Yes" or "No," depending on whether the provider is a non-profit organization.

ProviderAttrGet$(<provider.sub.-- num>, <attr.sub.-- name>)

Gets the value of the attribute named <attr.sub.-- name> for the content provider whose provider identifier is <provider.sub.-- num>. The value is returned as a string, but can be converted to any other appropriate type using the data type conversion functions provided by the Computation Language and the Script Language. Text to data conversion functions are well known in the art and are not discussed in this document.

UserTotalAccessCount%(<user.sub.-- num>)

Returns the total number of files that have ever been accessed by the user

whose user identifier is <user.sub.-- num>. The UserTotalAccessCountFunction% is useful for computing quantity discounts.

UserTotalAccessSize%(<user.sub.-- num>)

Returns the total size of the files that have ever been accessed by the user whose user identifier is <user.sub.-- num>. The User Total Access Size Function is useful for computing quantity discounts.

UserAttrSet(<user.sub.-- num>, <attr.sub.-- name>, <value>)

Associates, with the user whose user identifier is <user.sub.-- num>, an attribute name <attr.sub.-- name> having value <value>. If that attribute already exists for that user, replaces the value of the attribute with this new value. One example of using attributes on users might be to use an "Age" attribute to record the age of the user. This information might be used to offer senior citizen discounts on downloading fees, for example.

UserAttrGet$(<user.sub.-- num>, <attr.sub.-- name>)

Gets the value of the attribute named <attr.sub.-- name> for the user whose user identifier is <user.sub.-- num>. The value is returned as a string, but can be converted to any other appropriate type using the data type conversion functions provided by the Computation Language and the Script Language.

UserSearchTime#(<user.sub.-- num>, <provider.sub.-- num>)

Returns the total amount of time that the user whose user identifier is <user.sub.-- num> has been searching the content databases of the content provider whose provider identifier is <provider.sub.-- num>, in this session.

EntryCategoryCount%(<path>)

In a "yellow pages" or classified advertisement style service, returns the total number of categories under which the entry, whose file path name is <path>, has been listed. Entries that are listed under many categories can be charged a higher fee than entries that are listed in only a few categories.

ServerLoad!()

Returns the current load on the server as a value between 0.00 and 1.00, with 0.00 meaning no server load and 1.00 meaning that the server is fully loaded. The ServerLoad function can be used to set fees depending on the current load on the server. To discourage access during peak usage periods, higher prices can be assigned during peak usage times.

The Metering Subtool

A wide variety of metering capabilities are provided by the Molisa online service platform. The metering capabilities track the usage patterns of an online service, and the usage by users and other services. The metering information can provide invaluable feedback on the volume and duration of access to documents, subservices, and the online service as a whole. Furthermore, the metering information is available from the Fee Computation language using defined functions such that fees can be based on user usage.

It would be an unnecessary performance burden for the server to gather all possible statistics on all possible online service entities. With the Metering Tool, the developer indicates specifically which statistics should be gathered, and on which parts of the online service. The online service server tracks service usage in the ways specified in the metering subtool. (The server also gathers the specific usage data required by the fees indicated in the Fee Setter subtool.)

The Metering subtool allows the developer to manipulate a list of Metering Specifiers. Each Metering Specifier is a pair consisting of: (1) an online service entity (document, hyperlink, subservice, service, etc.), and (2) the particular property of that entity that should be metered. The properties that can be metered include: number of users who access the entity, number of minutes that they use the entity, total number of times that the entity was accessed, number of times the entity was viewed vs. downloaded, number of times the entity was requested by another service, times of day that entity is accessed, etc.

After gathering metering information, the online service provider can view the metering information in graph, chart, and numeric form, using separately provided analysis software. The metering information can be used to tune the performance of a server. For example, a developer can expand certain service areas that receive heavy use. Similarly, a developer can discard portions of an online service that are infrequently used, cost-justify a more powerful server to run the service, assess how often a user is "referred" to this service from another service, etc.

The Debugger Subtool

To facilitate the development of an online service, a Debugger subtool exists. The Debugger subtool provides a means for the developer to "run" an online service that is being developed. The Debugger subtool simulates an access to an online service from user client software such that a developer can test an online service by accessing the online service in the same manner that a user would,

The Debugger subtool, can be stopped at any point. When the Debugger is stopped, the developer can use an appropriate Utility Subtool to modify the currently displayed hyperdocument or subservice infrastructure (or any other part of the service). After modifying the subservice, the developer can resume the simulation of the online service from the stopping place. The Debugger also allows the developer to single-step through scripts, inspect and change script variables, and even modify the scripts themselves, like conventional debuggers for interpreted languages in other application domains.

Although the present invention has been described in terms of specific exemplary embodiments, it will be appreciated that various modifications and alterations might be made by those skilled in the art without departing from the spirit and scope of the invention as set forth in the following claims.

CLAIMS:

What is claimed is:

1. A system for developing an online service with a computer, comprising the elements of:

(a) a first editor for enabling a user to edit a data store that contains information comprising the online service;

(b) a second editor for enabling the user to define an interactive behavior of said online service, said second editor having a visual user interface for editing:

(i) a format in which said information from said data store of said online service is displayed;

(ii) a set of functional features provided by said online service; and

(iii) a set of visual objects for accessing said functional features, said set of visual objects being accessed by the user of the online service; and

(c) a fee setting tool, said fee setting tool for enabling the user to set fees associated with said online service for an entity.

2. The system of claim 1, wherein each of said fees are triggered by a defined user action.

3. The system of claim 2, wherein one of said defined user actions comprises access by said user to one of said visual objects on the online service.

4. The system of claim 2, wherein one of said defined user actions comprises submittal of an object for inclusion in said data store of said online service.

5. The system of claim 2, wherein one of said defined user actions comprises a traverse of a hyperlink.

6. The system of claim 2, wherein one of said defined user actions comprises a connection to the online service.

7. The system of claim 1, wherein each of said fees are triggered by passage of a defined amount of time.

8. The system of claim 1, wherein said fee setting tool defines a fee specifier for each fee, said fee specifier comprising a first field specifying a fee action that triggers said fee.

9. The system of claim 8, wherein said fee specifier further comprises a second field specifying an entity to whom the fee is directed.

10. The system of claim 8, wherein said fee specifier further comprises a third field specifying an object associated with said fee action.

11. The system of claim 8, wherein said fee specifier further comprises a fourth field defining a fee computation.

12. The system of claim 11, further comprising:

a script editor, said script editor for editing a script specifying said fee computation.

13. The system of claim 11, wherein said fourth field comprises a script specifying said fee computation.

14. The system of claim 13, wherein said script comprises at least one fee setting script primitive.

15. The system of claim 1, wherein said online service comprises a plurality of subservice programs.

16. The system of claim 1, wherein if said fee is positive, a fee is charged to said entity, and if said fee is negative, a payment is made to said entity.

17. The system of claim 1, wherein said online service distributes information using a Hyper Text Transport Protocol (HTTP), said information comprising a hypermedia document.

18. The system of claim 17, wherein said online service distributes information on the global Internet.

19. The system of claim 17, wherein said hypermedia document comprises a Hyper.sub.-- Text **Markup** Language (HTML) document.

20. A system for specifying fees for an entity associated with an online service comprising:

(a) means associated with an object of the online service for defining at least one of a plurality of triggering actions for a fee;

(b) means associated with a triggering action for defining a fee specification for the entity;

(c) means for editing a plurality of fee specifications for the entity; and

(d) means for storing the plurality of fee specifications using the editing means.

21. A system for determining a fee for an entity associated with an online service, comprising:

(a) means for detecting at least one of a plurality of actions on an object of the online service, said object being associated with an action;

(b) means, operative in response to detection of the action, for identifying a fee specification for the action and the object associated with the action;

(c) means for utilizing the fee specification to define the fee for the entity; and

(d) means for storing a plurality of fee specifications.

22. The system of claim 21, further comprising a fee specifier for specifying fees for the entity associated with the online service, comprising:

(a) means associated with an object of the online service for defining at least one of a plurality of triggering actions for a fee;

(b) means associated with the triggering action for defining a fee for the

entity;

(c) means for editing a plurality of fee specifications for the entity; and

(d) means for storing the plurality of fee specifications using the means for editing.

23. The system of claim 20, wherein each of said fees are triggered by a defined user action.

24. The system of claim 23, wherein one of said defined user actions comprises access by a user to one of said objects on said online system that is a visual object.

25. The system of claim 23, wherein one of said defined user actions comprises submittal of an object for inclusion in a data store of said online service.

26. The system of claim 23, wherein one of said defined user actions comprises a traverse of a hyperlink.

27. The system of claim 23, wherein one of said defined user actions comprises a connection to an online service.

28. The system of claim 20, wherein each of said fees are triggered by passage of a defined amount of time.

29. The system of claim 20 wherein said means for defining the fee specification define a fee specifier, wherein the fee specifier comprises a first field specifying a triggering action that triggers said fee.

30. The system of claim 29, wherein said fee specifier further comprises a second field specifying an entity to whom the fee is directed.

31. The system of claim 29, wherein said fee specifier further comprises a third field specifying an object associated with said fee.

32. The system of claim 29, wherein said fee specifier further comprises a fourth field defining a fee computation formula.

33. The system of claim 32, further comprising:

a script editor, said script editor enabling editing a script specifying said fee computation.

34. The system of claim 32, wherein said fourth field comprises a script specifying said fee computation formula.

35. The system of claim 34, wherein said script comprises at least one fee setting script primitive.

36. The system of claim 20, wherein if said fee is positive, a fee is charged to an entity, and if said fee is negative, a payment is made to said entity.

37. The system of claim 20, wherein a document object includes a Hyper.sub.-- Text **Markup** Language (HTML) document.

38. A system for developing an online service with a computer, comprising:

(a) a first editing module for displaying and enabling editing of relationships among document objects of the online service;

(b) a second editing module for enabling editing of individual document objects of the online service;

(c) a mechanism for invoking the second editing module in response to selection of a document object in the first editing module;

(d) means associated with said document object of the online service for defining at least one of a plurality of triggering actions for a fee;

(e) means associated with a triggering action for defining a fee specification for an entity;

(f) means for editing the fee specification for the entity; and

(g) means for storing a plurality of fee specifications using at least one of said first editing module and said second editing module.

39. The system of claim 38, wherein the relationships among the document objects are hyperlinks between the document objects.

40. A system for developing an online service with a computer, comprising:

(a) a viewing module for displaying relationships among and enabling selection of document objects of the online service;

(b) an editing module for editing individual document objects of the online service;

(c) a linking mechanism for invoking the editing module in response to selection of a document object in the viewing module;

(d) means associated with said document object of the online service for defining at least one of a plurality of triggering actions for a fee;

(e) means associated with a triggering action for defining a fee specification for an entity;

(f) means for editing the fee specification for the entity; and

(g) means for storing a plurality of fee specifications using the viewing module and the editing module.

41. A system for editing fee structures of an online service with a computer, comprising:

(a) means for displaying a visual representation of a fee specification having user-modifiable portions, wherein a user-modifiable portion is provided for entry of an indication of a document object of the online service, an indication of an event in connection with the document object, and a fee formula;

(b) means for receiving user input to edit a fee specification using the visual representation and for storing edited fee specifications; and

(c) means for storing a plurality of fee specifications defined using the means for displaying and means for receiving.

42. The system of claim 41, wherein the visual representation is a template.

43. The system of claim 41, wherein the fee formula is defined using a scripting language.

44. The system of claim 41, wherein the plurality of fee specifications are stored in a list.

Need

Reference

US-PAT-NO: 5740252
DOCUMENT-IDENTIFIER: US 5740252 A

TITLE: Apparatus and method for passing private demographic information between hyperlink destinations
DATE-ISSUED: April 14, 1998

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	RULE 47
Minor; Halsey M.	San Francisco	CA	N/A	N/A	N/A
Gandhi; Munish	Bridgewater	NJ	N/A	N/A	N/A
Headapohl; William E.	San Francisco	CA	N/A	N/A	N/A
Rosenberg; Jonathan	Annandale	NJ	N/A	N/A	N/A

ASSIGNEE INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	TYPE CODE
C/NET, Inc.	San Francisco	CA	N/A	N/A	02

APPL-NO: 8/ 542564
DATE FILED: October 13, 1995

INT-CL: [6] H04L009/00,H04L009/32

US-CL-ISSUED: 380/49,380/9 ,380/21 ,380/23 ,380/25
US-CL-CURRENT: 713/153,707/501 ,713/162
FIELD-OF-SEARCH: 380/4;380/9 ;380/21 ;380/23 ;380/25 ;380/49 ;380/50 ;380/59

REF-CIT ED:

U. S. PATENT DOCUMENTS

PAT-NO	ISSUE-DATE	PATENTEE-NAME	US-CL
5416842	May 1995	Aziz	N/A

OTHER PUBLICATIONS

Copy of U.S. Census Bureau Internet Homepage And Awards Page at http:// WWW.CENSUS.gov/, No Date.

Copy of "U.S. Export Restricted Software Download FAQ" At http://WWWUS.Netscape.com-/1Eng/US-Current/faq.html, No Date.

ART-UNIT: 222

PRIMARY-EXAMINER: Gregory; Bernarr E.

ATTY-AGENT-FIRM: Flehr Hohbach Test Albritton & Herbert LLP Galliani; William S. Hartnett; Clare T.

ABSTRACT:
A method of passing demographic information between computers includes the step of associating a computer operator with a set of demographic information. When the computer operator requests a transfer to another computer, a remote site information repository is accessed to identify a remote site destination address and a remote site encryption key. The demographic information is then encrypted using the remote site encryption key to form an encoded demographic signal. The encoded demographic signal is then combined with the remote site destination address to form an encoded demographic hyperlink transfer request, which is used to access the remote computer specified by the remote site destination address. The remote computer decrypts the encoded demographic information. The demographic information can be used by the remote computer in a number of ways, including the generation of a customized reply to an inquiry from the computer operator.

21 Claims, 6 Drawing figures
Exemplary Claim Number: 7
Number of Drawing Sheets: 5

BRIEF SUMMARY:
BRIEF DESCRIPTION OF THE INVENTION

This invention relates generally to hyperlink connections on computer networks. More particularly, this invention relates to a technique for appending private demographic information to a hyperlink transfer request used to move from one hyperlink computer to another hyperlink computer of a computer network.

BACKGROUND OF THE INVENTION

The term "hypertext" is used to describe documents with links to other documents. For example, a first document may include the word "Keats"; this word could then have a link to another document with the title "English Poets". These links are referred to herein as "hyperlinks". Hyperlinks are used between computers in computer networks.

A notable use of hyperlinks is on the World Wide Web (WWW) of computers. In this context, the operator of an end-user computer uses a "browser" to view a published page received from a first web computer. The published page may have a "hyperlink hot spot", which, when activated, results in a call to a second web computer. More particularly, the published page may have the hyperlink hot spot "Keats", which, when activated, causes the transfer of control to a

specified university computer with a file providing information on "English Poets".

Thus, each hyperlink hot spot knows the name of its associated file and on which computer that file is stored. The name of the computer and the file are combined into a Uniform Resource Locator (URL). A typical URL is http://SU/123. This URL is an instruction to retrieve the file 123 from the State University computer "SU" using a method called Hypertext Transport Protocol "HTTP". A URL may also be used to invoke a specified function on a remote computer, with the remote computer returning the results of the invoked and executed function.

Thus, when the hyperlink hot spot is selected, control is transferred across the WWW to the computer specified in the URL instruction. The computer then returns to the end-user computer the file specified in the URL instruction. Subsequently, the browser on the end-user computer is used to display the information associated with the file.

It is known in the art to use demographic information so that a reply to an inquiry from an end-user computer is tailored to the computer operator's interests. This technique is especially relevant in a commercial setting where a computer operator is accessing different computers to identify products of interest. For example, a computer operator may have relied upon hyperlink hot spots to connect to an on-line computer catalog. In this context, it can be appreciated that the owner of the on-line computer catalog would like to have demographic information on the computer operator so that the information sent to the computer operator is particularly relevant. For example, the information should only relate to Apple architecture computers and peripherals if the relevant demographic information specifies that this is the type of system that the computer operator uses.

One way to obtain demographic information is for a computer operator to register the demographic information at each web site that is visited. The obvious problem with this approach is that a user does not want to take the time to repeatedly provide demographic information.

A more sophisticated technique of utilizing demographic information is shown in FIG. 1. FIG. 1 illustrates a technique in which an operator of an end-user computer 20 registers demographic information at a web site demographic database 24A. After the registration process is completed, the web site demographic database 24A passes a unique identifier to the end-user computer 20. Subsequently, when the end-user computer 20 requests information from another computer on the WWW, say web site B, the unique identifier is passed to the computer with the request. In response to the request, the web site 24B establishes a communication link with the demographic database web site 24A.

The unique identifier is then passed from the web site 24B to the demographic database website 24A. The demographic database website 24A uses the unique identifier as an address to locate the demographic information associated with the computer operator of end-user computer 20. This information is then passed back to the web site 24B. The web site 24B, relying upon the demographic information, then supplies a customized reply to the original request from the end-user computer 20.

There are a number of problems associated with the system of FIG. 1. First, the operator of the end-user computer 20 must pass a unique identifier to each web site that is called. Consequently, without the knowledge of the computer operator, it is possible to coordinate every web site that a computer operator visits. This potentially provides a great deal of information regarding the computer operator and his or her interests. Moreover, the demographic information about the individual is passed over communication links where it can be accessed, thereby providing more information on the computer operator. Most individuals view each of these scenarios as a potential invasion of privacy.

Another problem with the system of FIG. 1 is that a communication link must be established with the demographic database web site 24A each time a new web site is visited. Consequently, the system of FIG. 1 can be relatively slow.

In view of the foregoing, it would be highly advantageous to provide a simple technique for efficiently passing private demographic information between hyperlink destinations in a computer network.

SUMMARY OF THE INVENTION

The method of the invention passes demographic information between computers by associating a computer operator with a set of demographic information. When the computer operator requests a transfer to another computer, a remote site information repository is accessed to identify a remote site destination address and a remote site encryption key. The demographic information is then encrypted using the remote site encryption key to form an encoded demographic signal. The encoded demographic signal is then combined with the remote site destination address to form an encoded demographic hyperlink transfer request, which is used to access the remote computer specified by the remote site destination address. The remote computer decrypts the encoded demographic information.

There are a number of benefits associated with this technology. First, computers in a network are able to receive and process demographic information. This is achieved without requiting the computer operator to register at each site on the network. In addition, the demographic information is encoded when

it is on public transmission channels. Thus, the information remains private. The encoding operation also promotes privacy to the extent that the information can only be processed by a computer that can decode the information.

The disclosed system avoids several of the pitfalls associated with the prior art system of FIG. 1. First, the disclosed system does not rely upon a unique identifier that can be used to trace the different locations that a computer operator visits. In addition, since demographic information is directly passed to a computer in the present invention, a separate communication session need not be initiated with a demographic database to obtain the information.

DRAWING DESCRIPTION:

BRIEF DESCRIPTION OF THE DRAWINGS

For a better understanding of the nature and objects of the invention, reference should be made to the following detailed description taken in conjunction with the accompanying drawings, in which:

FIG. 1 illustrates a prior art system that uses demographic information to create a customized reply to a request for information.

FIG. 2 illustrates a system for appending demographic information to a hyperlink transfer request, in accordance with one embodiment of the invention.

FIG. 3 is a more detailed illustration of the processing associated with the system of FIG. 2.

FIG. 4 illustrates a computer system configuration that may be used in accordance with the invention.

FIG. 5 illustrates the processing associated with the demographic information encoding and decoding programs used in accordance with an embodiment of the invention.

FIG. 6 is a detailed illustration of alternate processing steps that may be used in accordance with the system of FIG. 2.

Like reference numerals refer to corresponding parts throughout the several views of the drawings.

DETAILED DESCRIPTION:

DETAILED DESCRIPTION OF THE INVENTION

FIG. 2 is a general illustration showing the architecture and methodology of

the system of the present invention. The system architecture includes an end-user computer 30 that generates a request that reaches an entry web site 32A. The request uses a Uniform Resource Locator (URL) address to specify a computer (entry web site 32A) and information (a file) at that computer.

The entry web site 32A is used to obtain demographic information about the computer operator at the end-user computer 30. That is, the entry web site 32A provides a set of inquiries or a graphical interface to obtain demographic information about the computer operator. Typical demographic information includes name, age, zip code, income level, employment information, type of computer used, other purchasing preferences, etc. This session also prompts the computer operator to select a password for use in future communications. The demographic information is then stored in a database at the entry web site 32A. In subsequent accesses to the entry web site 32A, the computer operator need not register again, instead the computer operator is recognized by providing authenticating information, such as a name and password.

After authentication or an initial registration session, the entry web site 32A provides a reply to the original inquiry. That is, the entry web site 32A passes back to the end-user computer 30 the file specified by the URL command. A browser running on the end-user computer 30 is used to display the file.

When the computer operator views the received file he or she may activate a hyperlink hot spot in the displayed document. This causes a new request 16 by sent to the entry web site 32A. The request is a hyperlink transfer request to a new computer, say remote web site 34A. The entry web site 32A correlates the request with the computer operator and then fetches the previously accumulated demographic information regarding the computer operator. The entry web site 32A also accesses a remote site information repository stored within the entry web site 32A. The remote site information repository includes a list of remote site destination addresses and corresponding remote site encryption keys. The remote site destination addresses include the URL addresses for the different computers within the system that may receive demographic information. Each remote site destination address has a corresponding remote site encryption key. The encryption key is used to encode the demographic information that is sent to a computer. When the information is received at the computer, the inverse key is used at the computer to decrypt the demographic information.

Thus, the reply from the entry web site 32A to the end-user computer 30 is a remote site destination address and demographic information encrypted by the selected remote site encryption key. This response is referred to as a demographic hyperlink transfer request. As shown in FIG. 2, the end-user computer 30 passes the returned demographic hyperlink transfer request to a remote web site 34A. The remote web site 34A has the address specified by the remote site destination address within the encoded demographic hyperlink

transfer request.

The remote web site 34A is an associated web site in the system of FIG. 2. It is associated in the sense that it can process the demographic information that is passed to it. As indicated above, the entry web site 32A stores the address for each associated computer in the system.

The remote web site 34A uses the inverse of the same encryption key specified at the entry web site 32A to decrypt the received demographic information. That is, since it is an associated computer in the system of FIG. 2, it has a specified encryption key that is also stored at the entry web site 32A. Thus, when the remote web site 34A receives information, it can use the encryption key to decrypt the information that was encoded at the entry web site 32A.

Subsequently, the demographic information may be used to supply a demographic-tailored reply to the end-user computer 30. A demographic-tailored reply is a reply to a request from another computer that relies upon at least one piece of demographic information supplied by an encoded demographic hyperlink transfer request. The demographic-tailored reply relies upon this information by providing information that is consistent with the interests of an individual with the specified demographic information. The demographic information may also be used to obtain a profile of the type of computer operators that are accessing a particular computer.

In view of the foregoing general description of the invention, those skilled in the art will recognize a number of benefits associated with the disclosed technology. First, computers in a network are able to receive and process demographic information. This is achieved without requiting the computer operator to register at each site on the network. In addition, the demographic information is encoded when it is on public transmission channels. Thus, the information remains private. The encoding operation also promotes privacy to the extent that the information can only be processed by a computer that can decode the information. Thus, computers that are not registered at the entry web site 32A cannot process demographic information.

The disclosed system avoids several of the pitfalls associated with the prior art system of FIG. 1. First, the disclosed system does not rely upon a unique identifier that can be used to trace the different locations that a computer operator visits. In addition, since demographic information is directly passed to a computer, a separate communication session need not be initiated with a demographic database to obtain the information.

The general nature and benefits of the invention have now been disclosed. Attention presently turns to a more detailed discussion of the system of the

invention. FIG. 3 illustrates end-user computer 30, entry web site 32A, and remote web site 34A. In particular, the figure shows the processing steps that are performed at each location. It should be appreciated that a system in accordance with the invention may have a large number of end-user computers 30, entry web sites 32A (say 32A-32N), and remote web sites 34A (say 34A-34N).

An entry web site 32 refers to a site at which demographic information is received. Otherwise, an entry web site 32 can be considered to be interchangeable with the remote web sites 34. That is, in a preferred embodiment, the entry web site 32 passes and receives encoded demographic information, just any other remote web site 34 in the system.

The first processing step shown in FIG. 3 is to make a connection (step 40) with the entry web site 32A. This is a standard connection step between an end-user computer 30 and a server computer 32. The first operation performed on the entry web site 32A is to authenticate the user with a name and password (step 42). This step may include a registration process, as discussed above. However, in this example, it will be assumed that the computer operator has already visited the entry web site 32A and therefore only an authentication operation is required.

The next operation performed by the computer 32A is to generate a reply (step 44). The reply is the file or other information requested from the computer 32A by the end-user computer 30. The reply is passed back to the end-user computer 30, where it is viewed (step 46). The reply, typically one or more pages displayed on a browser, includes one or more hyperlink hotspots. Activating a hyperlink hotspot causes a return of control to the entry web site machine 32A.

At this point, the entry web site machine 32A knows that there has been a request to access another computer. Thus, operations are performed so that when control is passed to the other computer, demographic information regarding the computer operator is available to the other computer. Specifically, a correlation operation is performed to match the computer operator with his or her corresponding demographic information (step 48). This correlation operation is executed by relying upon the information obtained during the authentication operation (block 42).

The next processing step is to encrypt the demographic information (step 50). The demographic information is encrypted by relying upon a remote site encryption key that is stored with a remote site destination address in a remote site information repository. Thus, the remote site information repository includes an address for each computer in the system and an encryption key for each address.

After the demographic information is encrypted with the retrieved encryption key, a transparent hyperlink request is sent to the end-user computer 30. The transparent hyperlink request includes the remote site destination address and the encoded demographic information. The end-user computer 30 automatically, without computer operator participation, processes the remote site destination address by calling the remote site computer specified by the remote site destination address, in this case, remote web site 34A. This automatic passing of information from a first server (32A), to an end-user computer (30), and to a second server (34A) is sometimes referred to as a "redirect".

The remote web site 34A decrypts the demographic information using the encryption key that it stores locally (step 54). Thus, the same encryption key is stored at the remote web site 34A and in the remote site information repository of entry web site 32A. The decrypted demographic information can be used in any number of ways. For instance, it can be used to gather profile information on computer operators accessing the remote web site 34A. FIG. 3 illustrates that the information is used to generate a demographic-tailored reply (step 56). As shown in FIG. 3, this reply is passed back to the end-user computer 30, where it is viewed by the computer operator (block 58).

FIG. 4 illustrates the hardware elements that are used in practicing the invention. In the figure, it is seen that the end-user computer 30 includes a Central Processing Unit (CPU) 60 which is attached to a memory 62. The memory 62 stores a set of executable programs, including a browser program 64. As known in the art, the browser program 64 is used to access, communicate, and display information from other computers. The other computers are accessed through a network connection 66, which is linked to a transmission channel 70. The transmission channel 70 may be any wire or wireless communication network. The entry web site 32A is connected to the transmission channel 70. The entry web site 32A includes a network connection 72, which communicates with a CPU 74. The CPU 74 is connected to a memory 76, which stores a set of executable programs and data. The memory 76 stores a communication program 78 that is used to communicate with other computers connected to the transmission channel 70. The memory 76 also stores demographic information 80 on the different computer operators that have registered at the computer 32A. A demographic information encoding program 82 and a remote site information repository 84 are also stored in the memory 76. As will be described below, the demographic information encoding program 82 accesses the remote site information repository 84 for remote site destination addresses and remote site encryption keys. The program then uses this information to encrypt the demographic information 80 stored on a particular computer operator.

The system of FIG. 4 also includes a remote web site computer 34A. The remote web site computer 34A includes a network connection 90, a CPU 92, and a memory 94. The memory 94 stores a communication program 96 of the type

previously described. In addition, the memory 94 stores a demographic information decoding program 98. This program uses the same encryption key and an inverse encryption operation to decode the incoming encrypted information. The remote web site computer 34A may also include a page customization program. As its name implies, this program uses the decoded demographic information to create a demographic-tailored reply to the incoming request for information.

Turning now to FIG. 5, depicted therein is a more detailed representation of the processing performed by the entry web site 32A and remote web site 34A. The first step associated with this processing is that a hyperlink transfer request is obtained from a computer operator (step 110) at an end-user computer 30. For example, assume that a computer operator, U, at end-user computer 30 has previously been authenticated at an entry web site 32A called "cnet". Thereafter, the computer operator views, at the end-user computer 30, a page, provided by cnet, containing ads for computer peripherals. Each ad is associated with a different computer vendor running a separate server (remote web site). The computer operator then decides to obtain information from one of those servers, say the Internet Shopping Network (ISN), by clicking on the hyperlink hot spot for this location. This selection causes the following request: "http://www.cnet.com/cgi-bin/goto?isn" to be sent from the end-user computer 30 over the transmission channel 70 to the entry web site 32A (the cnet computer).

The entry web site 32A then uses the information received during the authentication operation to correlate demographic information with the computer operator (step 112). That is, demographic information, referred to as D, is correlated with the computer operator U.

Subsequently, the remote site information repository 84 is accessed (step 114) to identify the remote site destination address corresponding to the "isn" string. By way of example, this location is specified as "www.isn.com".

The repository 84 is also accessed (step 114) for the remote site encryption key (referred to as "K") corresponding to the "isn" string. Standard techniques are used to encode the demographic information using the remote site encryption key (step 116). This results in an encoded demographic signal segment, referred to as K(D). The encoded demographic signal segment is combined with the remote site destination address (step 118) to form an encoded demographic hyperlink transfer request, for example, of the following form: "Location:http://www.isn.com/new-user?K(D)".

The end-user computer 30 receives the encoded demographic hyperlink transfer request. The first term in the request is the word "Location" so the end-user computer 30 knows that a new location has been specified. The request also indicates that the "http" protocol should be used to connect with the computer

"www.isn.com". With this information, standard techniques are used to access the remote web site 34A. At this site, processing is passed to the demographic information decoding program 98. The program 98 identifies the "new-user?" designation as a command to process the encoded demographic information "K(D)". The question mark symbol "?" is used in the HTTP protocol to indicate that additional information is appended to a command. In this case, the information is the encoded demographic information "K(D)".

The demographic information decoding program 98 decrypts the demographic information (step 120) by using the inverse of K to decrypt the information "D". This results in demographic information in a predetermined format. For example, the predetermined format may specify that the first bytes define a zip code, the next byte defines an age, etc. The demographic information can then be used to create a demographic-tailored reply (step 122) using the page customization program 100. The page customization program 100 may incorporate any set of instructions to process the demographic information, and in response thereto, select information for the tailored reply. The remote web site 34A then passes the demographic-tailored reply to the computer operator (step 124) at end-user computer 30.

FIG. 6 illustrates the processing associated with an alternate embodiment of the invention. In particular, FIG. 6 illustrates that each reply sent to an end-user computer 30 may include a page with all the information necessary to compute a hyperlink transfer request at the end-user computer 30. That is, the end-user computer 30 receives a page with a set of information embedded therein. The embedded information includes the remote site destination address for each hyperlink hot spot appearing in the page. In addition, the remote site destination address has a corresponding remote site encryption key to encrypt the demographic information which is also embedded into the page. Consequently, when a hyperlink hot spot is activated, an encoded demographic transfer request may be routed from the end-user computer 30, without returning to the entry web site 32A. The problem with this approach is that for each hyperlink hot spot in a page, the remote site destination address and remote site encryption key must be fetched. This requires more processing time. Moreover, since only one hyperlink hot spot can be activated on a page, information is gathered, but never used.

It will be clear to those skilled in the art that a variety of alternate configurations may be used in accordance with the disclosed technology. For instance, it is not important whether the remote site destination address is obtained before correlating the computer operator with the demographic information. In addition, the designation between an entry web site 32 and a remote web site 34 is somewhat arbitrary. An entry web site 32A may include a demographic information decoding program 98 and a page customization program 100. Similarly, it should be appreciated that a remote web site 34A may have a

remote site information repository 84 and a demographic information encoding program 82 so that it can reconvey demographic information that it has received.

The foregoing descriptions of specific embodiments of the present invention are presented for purposes of illustration and description. They are not intended to be exhaustive or to limit the invention to the precise forms disclosed, obviously many modifications and variations are possible in view of the above teachings. The embodiments were chosen and described in order to best explain the principles of the invention and its practical applications, to thereby enable others skilled in the art to best utilize the invention and various embodiments with various modifications as are suited to the particular use contemplated. It is intended that the scope of the invention be defined by the following claims and their equivalents.

US-PAT-NO: 6167253
DOCUMENT-IDENTIFIER: US 6167253 A

TITLE: **Mobile** data/message/electronic mail download system utilizing network-centric protocol such as Java
DATE-ISSUED: December 26, 2000

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	RULE 47
Farris; Robert D.	Sterling	VA	N/A	N/A	N/A
Flaherty; Stephen J.	Upper Marlboro	MD	N/A	N/A	N/A
Goodman; William D.	Collegeville	PA	N/A	N/A	N/A

ASSIGNEE INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	TYPE CODE
Bell Atlantic Network Services, Inc.	Arlington	VA	N/A	N/A	02

APPL-NO: 8/ 998020
DATE FILED: December 24, 1997

PARENT-CASE:
RELATED APPLICATIONS This application is a continuation-in-part application from copending U.S. patent application Ser. No. 08/9114,234 filed on Nov. 4, 1997, from copending U.S. patent application Ser. No. 08/923,122 filed on Sep. 4, 1997, which is a continuation-in-part application from copending U.S. patent application Ser. No. 08/725,385 filed on Oct. 3, 1996, now U.S. Pat. No. 5,694,455 which is a continuation of copending U.S. patent application Ser. No. 08/371,902 filed on Jan. 12, 1995, now U.S. Pat. No. 5,594,779, all of which are hereby incorporated by reference.

INT-CL: [7] H04M011/10

US-CL-ISSUED: 455/412,455/413 ,455/414
US-CL-CURRENT: 455/412,455/413 ,455/414
FIELD-OF-SEARCH: 455/413;455/412 ;455/414 ;455/418 ;455/419 ;455/466 ;455/557 ;455/4.1 ;455/4.2 ;455/6.3 ;370/328 ;370/352 ;709/246 ;709/227 ;709/228 ;709/230 ;709/236

REF-CITED:

U.S. PATENT DOCUMENTS

PAT-NO	ISSUE-DATE	PATENTEE-NAME	US-CL
4124773	November 1978	Elkins	N/A

4481382	November 1984	Villa-Real	N/A
4577067	March 1986	Levy et al.	N/A
4599490	July 1986	Cornell et al.	N/A
4750198	June 1988	Harper	N/A
4754495	June 1988	Kawano et al.	N/A
4799253	January 1989	Stern et al.	N/A
4812843	March 1989	Champion, III et al.	N/A
4849811	July 1989	Kleinerman	N/A
4893327	January 1990	Stern et al.	N/A
4905301	February 1990	Krolopp et al.	N/A
4922518	May 1990	Gordon et al.	N/A
4932046	June 1990	Katz et al.	N/A
4954958	September 1990	Savage et al.	N/A
4972455	November 1990	Phillips et al.	N/A
5008925	April 1991	Pireh	N/A
5010399	April 1991	Goodman	N/A
5020091	May 1991	Krolopp et al.	N/A
5020092	May 1991	Phillips et al.	N/A
5020093	May 1991	Pireh	N/A
5105197	April 1992	Claggett	N/A
5111534	May 1992	Benner	N/A
5119397	June 1992	Dahlin et al.	N/A
5119502	June 1992	Kallin et al.	N/A
5121126	June 1992	Clagett	342/419
5131020	July 1992	Liebesny et al.	455/413
5148471	September 1992	Metroka et al.	N/A
5177780	January 1993	Kasper et al.	N/A
5206641	April 1993	Grant et al.	340/905
5218629	June 1993	Dumond, Jr. et al.	N/A
5222120	June 1993	McLeod et al.	N/A
5243640	September 1993	Hadley et al.	N/A
5247347	September 1993	Litteral et al.	348/7
5247698	September 1993	Sawyer et al.	N/A
5251249	October 1993	Allen et al.	N/A
5257400	October 1993	Yoshida	N/A
5307400	April 1994	Sawyer et al.	N/A
5313515	May 1994	Allen et al.	N/A
5319641	June 1994	Fridrich et al.	N/A
5329578	July 1994	Brennan et al.	N/A
5353287	October 1994	Kuddes et al.	N/A
5353331	October 1994	Emery et al.	N/A
5353352	October 1994	Dent et al.	N/A
5369681	November 1994	Boudreau et al.	N/A
5371898	December 1994	Grube et al.	N/A

5418784	May 1995	Ramakrishnan et al.	N/A
5436960	July 1995	Campana, Jr. et al.	455/412
5440336	August 1995	Buhro et al.	348/13
5444768	August 1995	Lemaire et al.	379/68
5457732	October 1995	Goldberg	N/A
5487101	January 1996	Fletcher	455/435
5506887	April 1996	Emery et al.	N/A
5559860	September 1996	Mizikovsky	N/A
5594779	January 1997	Goodman	455/413
5692032	November 1997	Seppanen et al.	455/466
5694455	December 1997	Goodman	455/413
5727159	March 1998	Kikinis	709/246
5745551	April 1998	Strauch et al.	455/412
5802299	November 1998	Logan et al.	709/218
5841979	November 1998	Shulhof et al.	709/237

FOREIGN PATENT DOCUMENTS

COUNTRY	FOREIGN-PAT-NO	PUBN-DATE	US-CL
GB	2216319	October 1989	

OTHER PUBLICATIONS

"**Wireless** Web browsing: How long will deployment take?", by Arielle Emmett (Dec. 15, 1996), **Wireless** Web Browsing, pp. 1-5.

"Tomorrow's Internet is Here Today", Smith B., Computer, Apr. 1997, pp. 22 and 23.

"Communications", Bell & Riezenman, IEEE Spectrum, Jan. 1997, pp. 27-37.

"Java and Beyond: Executable Content", K. Srinivas et al., Computer, Jun. 1997, pp. 49-52.

"Cafe: The right place for Java", D. Krishna, Computer, Jul. 1996, pp. 104-108.

"Java, the Web, and Software Development", E. Yourdon, Computer, Aug. 1996, pp. 25-30.

"The Case for Java as a Programming Language", A. van Hoff, IEEE Internet Computing, Jan./Feb. 1997, pp. 51-56.

"Software Engineering", Comerford, IEEE Spectrum, Jan. 1997, pp. 65-69.

"Java and the Shift to Net-Centric Computing", M. Hamilton, Computer, Aug.

1996, pp. 31-39.

"Enterprise Computing: The Java Factor", P. Ram et al., Computer, Jun. 1997, pp. 83-87.

"Java- and CORBA-Based Network Management", M. Leppinen et al., Computer, Jun. 1997, pp. 83-87.

"The Feel of Java", J. Gosling, Computer, Jun. 1997, pp. 53-57.

"If Java Is the Answer, What Was the Question?", T. Lewis, Computer, Mar. 1997, pp. 136, 133-135.

ART-UNIT: 274

PRIMARY-EXAMINER: Trost; William G.

ASSISTANT-EXAMINER: Redmon; Joy

ATTY-AGENT-FIRM: McDermott, Will & Emery

ABSTRACT:

In an information distribution system having **mobile** users, a method of distributing communication channels to **mobile** users includes transmitting information signals including at least one of Web related information and Internet related information received via one ore more of the Internet, ADSL, another **mobile,** a land-based user, and the at least one information service provider. The method also includes receiving the information signals from one or more of the Internet, ADSL, another **mobile,** a land-based user, and the information service provider in the receiver of the **mobile** terminal, and storing the information signals in its entirety in the **mobile** terminal prior to broadcasting and/or displaying same to the **mobile** user. The method also formats, broadcasts and/or displays the information signals after being stored in the **mobile** terminal.

47 Claims, 37 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 37

BRIEF SUMMARY:

BACKGROUND OF THE INVENTION

1. Technical Field

The present invention relates to a **mobile** data/message/electronic mail

download system utilizing network-centric protocol such as Java, and more particularly, to a data/message/electronic mail download system architecture for receiving data/message/electronic mail messages using a network-centric protocol such as Java from external systems, e.g., information service providers, using a **mobile, wireless,** digital and/or **cellular** telephone or transceiver system.

2. Background Art

Distribution of audio information or data has evolved from early radio broadcasting to meet viewer demand. Initially, radio receivers were bulky and essentially non-movable units which were located in the living room of a home, as a permanent fixture. The radio receiver has been significantly improved over the years to become more portable and convenient. For example, today most vehicles include radio receivers which broadcast prescheduled audio programming to the passengers of the vehicle. In addition, the radio receiver has been reduced to such a small size that many people keep such a portable device on their person while walking or exercising to enhance activities which were commonly performed without the convenience or entertainment of the radio receiver.

However, audio programming was, at best, prescheduled and typically randomized, with the listener having to tune to the designated station or frequency at the appointed time to listen to a particular audio program. Thus, audio listeners were subjected to the selection chosen by the particular broadcast station.

Technological advances resulted in the proliferation of Audio Cassette Recorders (ACR) and Video Cassette Recorders (VCR), establishing a second option for audio and video programming distribution. Pre-recorded audio and video programs are now available for sale and rental to ACR and VCR owners. Using an ACR or VCR, the viewer selects from among many titles available for sale and rental, and listens and perhaps views the program when convenient. The ACR or VCR owner further has the capability to selectively listen or view the programming using special functions of the ACR or VCR, such as pause, fast forward, reverse, slow motion, etc. The listener or viewer can thus manipulate and replay portions of the program at will.

The penalty for this convenience, however, is in the necessity to travel to the local rental/sales store, if necessary wait for a popular program tape to become available, and once the program is purchased or rented to return home to listen to the program. If the tape is rented, the listener then revisits the video store to return the tape.

Much research has been conducted in the unrelated arena of cable television

network programming. For example, cable television systems have developed and distributed "premium" channels viewable only by subscribers having appropriate descramblers. The descramblers are tuned to receive these premium channels, descramble the video and audio information and supply a signal capable of reception on a standard television set.

Pay-per-view programs, which evolved later, include recently released movies, live concerts and popular sporting events. Subscribers wishing to view a pay-per-view program place an order with the cable operator. At the designated time, the subscriber's descrambler is activated to permit viewing of the pay-per-view programming. However, the subscriber is restricted to viewing the programming at the scheduled time. There is no capability of delivering programming, video or audio, to a subscriber on demand, that is, immediately or at a subscriber-specified time and date. Further, these cable television systems provide the requested pay-per-view program at a stationary television with a stationary converter descrambler using stationary land lines.

Telephone lines have been suggested as an alternative means of video distribution in Goodman et al., U.S. Pat. No. 5,010,319 and Kleinerman, U.S. Pat. No. 4,849,811. However, systems using the public switched telephone network (PSTN) are often bandwidth limited, providing only still frame or video conferencing capabilities. Because telephone system carriers for the most part use the PSTN only for connectivity between subscribers, there is no capability for dynamic routing of digitized video without dedicated leased, wide bandwidth circuits. Telephone line based systems also fail to provide acceptable VCR type functional control of the programming.

U.S. Pat. No. 5,247,347, to Litteral et al., incorporated herein by reference, describes a so-called Video-on-Demand service that provides video programming to subscribers over the PSTN. A video information provider (VIP) transmits coded digital video data over wideband PSTN supplied connectivity to a central office. The video data may be buffered at the central office for transmission over a POTS line to the subscriber. A subscriber may use either a standard telephone instrument over the PSTN or a dedicated control device over an ISDN packet network to order the video programming. Such a device is located at a television set of the subscriber and permits a display of the program menu on the television screen.

Connectivity between the central office and the subscriber for transmission of video data is provided by an asymmetrical digital subscriber line (ADSL) system. ADSL interface units perform multiplexing of digital video information with voice information to be transmitted to the subscriber and support transmission on the ISDN packet data network of a reverse control channel from the subscriber to the central office.

However, all these prior art attempts have concentrated on video-on-demand programming which is tied to the public switched telephone network using stationary converted or digital subscriber devices located at a fixed location, such as the home.

U.S. Pat. No. 5,131,020 to Liebensy et al. describes a Method of and System for Providing Continually Updated Traffic or Other Information to Telephonically and Other Communications-Linked Customers. This patent pertains to a method of traffic information and telephone channel communication between a central station and a plurality of callers distributed in different zones throughout a geographical area. All callers are telephonically linked with the central station. The method collects and updates traffic information from a plurality of sources on a real-time and continual basis for all the zones throughout the area. It responds to telephone dialing on the caller's telephone keyboard and enters on such keyboard a code for the particular zone of interest specified by the caller. It telephonically transmits back from the central station to the caller a report of the traffic information requested by the caller in the particular zone specified by the caller. It also responds to subsequent caller keyboard requests for automatic updating of significant changes in the traffic information within such specific zone.

U.S. Pat. No. 5,243,640 to Hadley et al. relates to an Integrated **Cellular** Telephone and Vehicular Audio System. The patent pertains to interfacing a **mobile** telephone and an audio system in a vehicle. The patent integrates the two systems in order to share components and thereby eliminate duplication costs and complexity of the system. The system selectively couples program audio signals and phone audio signals to an output transducer depending on the activation of a main program audio system and telephone.

U.S. Pat. No. 5,206,641 to Grant et al. involves a Portable Traffic Congestion Radio. The patent pertains to a portable electronic storage device that receives and stores digitally coded traffic reports for a covered geographical area. The device presents traffic information relevant to a user-specified vehicle trip within the covered area. The device includes a touch-sensitive map that is used to indicate trip origin, destination and routing of interest. The device makes calculations to select and modify the reports and the traffic information from the reports is presented to the user by synthesized or digitized voice sounds.

U.S. Pat. No. 4,812,843 to Champion et al. relates to a Telephone Accessible Information System. The patent describes a communication system for subscribers that is capable of continuously updating information on a variety of subjects. Primarily, the patent deals with the subject of updated traffic information. Each geographic area served by the system is represented by a specially designed map. The map is divided into grid sections and systems to

indicate routes. The subscriber, through codes on a DTMF phone selects a particular route. The communications system, from information gathered in a database, provides the subscriber with updated traffic information. This is continually updated for a certain route for a certain period of time.

Heretofore, however, the prior art has not addressed the issue or problem relating to the providing of interactive information, data messages, **voice mail** and/or electronic mail messages to users which typically receive data/message/electronic mail messages using a network-centric protocol such as Java from external systems, e.g., information service providers, using a **mobile, wireless,** digital and/or **cellular** telephone or transceiver system.

In addition, the prior art has not considered or addressed the problem relating to the interactive selection of information, data messages, **voice mail** and/or electronic mail messages to users which typically receive data/message/electronic mail messages using a network-centric protocol such as Java from external systems where the information and/or messages are to be delivered or transmitted to a portable, moveable telephone-like device.

The prior art has further not addressed the problem providing the user of a portable audio device with an economical means of interactively receiving such information, data, **voice mail** and/or electronic mail messages.

The prior art has also not addressed the problem of efficiently allocating sufficient resources between the interactive relationship of the optional information provider and the portable audio program listener.

SUMMARY OF THE INVENTION

It is therefore, a feature and advantage of the present invention to provide data, **voice mail** and/or electronic mail to users which typically receive or listen to in a moveable or transient manner. An integral part of this feature and advantage of the present invention is that we have discovered that portable audio listeners have much different requirements than the typical video-on-demand systems. Accordingly, the present invention incorporates the considerations of the audio listeners in the overall architecture of the **mobile** audio program selection system of the present invention.

Another feature and advantage of the present invention is to provide the listener with data, **voice mail,** and/or electronic mail message selection from a voice/electronic mail provider where the audio messages are to be delivered or transmitted to a portable, moveable audio device.

Another feature and advantage of the present invention is to provide the user of a portable audio device with an economical means of receiving the data,

voice mail, electronic mail and/or audio programming.

Yet another feature and advantage of the present invention is to efficiently allocate sufficient resources between the data, electronic mail and/or **voice mail** program provider and the portable audio program listener.

Another feature and advantage of the present invention is in the providing of interactive information, data messages, **voice mail** and/or electronic mail messages to users which typically receive data/message/electronic mail messages using a network-centric protocol such as Java from external systems, e.g., information service providers, using a **mobile, wireless,** digital and/or **cellular** telephone or transceiver system.

Another feature and advantage of the present invention is to provide interactive selection of information, data messages, **voice mail** and/or electronic mail messages to users which typically receive data/message/electronic mail messages using a network-centric protocol such as Java from external systems, where the information and/or messages are to be delivered or transmitted to a portable, moveable telephone-like device.

Another feature and advantage of the present invention is to provide the user of a portable audio device with an economical means of interactively receiving such information, data, **voice mail** and/or electronic mail messages.

Another feature and advantage of the present invention is to efficiently allocate sufficient resources between the interactive relationship of the optional information provider and the portable audio program listener.

There has thus been outlined, rather broadly, the more important features of the invention in order that the detailed description thereof that follows may be better understood, and in order that the present contribution to the art may be better appreciated. There are, of course, additional features of the invention that will be described hereinafter and which will form the subject matter of the claims appended hereto.

In this respect, before explaining at least one embodiment of the invention in detail, it is to be understood that the invention is not limited in its application to the details of construction and to the arrangements of the components set forth in the following description or illustrated in the drawings. The invention is capable of other embodiments and of being practiced and carried out in various ways. Also, it is to be understood that the phraseology and terminology employed herein are for the purpose of description and should not be regarded as limiting.

As such, those skilled in the art will appreciate that the conception, upon

which this disclosure is based, may readily be utilized as a basis for the designing of other structures, methods and systems for carrying out the several purposes of the present invention. It is important, therefore, that the claims be regarded as including such equivalent constructions insofar as they do not depart from the spirit and scope of the present invention.

Further, the purpose of the foregoing abstract is to enable the U.S. Patent and Trademark Office and the public generally, and especially the scientists, engineers and practitioners in the art who are not familiar with patent or legal terms or phraseology, to determine quickly from a cursory inspection the nature and essence of the technical disclosure of the application. The abstract is neither intended to define the invention of the application, which is measured by the claims, nor is it intended to be limiting as to the scope of the invention in any way.

To achieve these features and advantages, the present invention provides a **mobile** audio program selection system. In one of the preferred embodiments, the **mobile** audio program selection system includes a radio frequency based information distribution system having **mobile** users. The distribution system includes a **mobile** switching office selectively connecting the **mobile** users of the information distribution system, and information service providers, operatively connected to the **mobile** switching office, at least one of the information service providers receiving user selection signal inputs received by the **mobile** switching office from a **mobile** user, and transmitting user selected information to the **mobile** switching office responsive to the user selection signal inputs. In addition, the distribution system includes at least one **mobile** terminal. The **mobile** terminal includes a receiver, operatively coupling the **mobile** terminal to the **mobile** switching office, and receiving the user selected information from the at least one of the information service providers via the **mobile** switching office. The **mobile** terminal also includes a control processor controlling operations of the at least one **mobile** terminal, and means for receiving a user selection from the **mobile** user, for generating user selection signal inputs responsive to the user selection, and for transmitting the user selection signal inputs to the **mobile** switching office. The **mobile** terminal further includes signal format means for formatting the user selected information for broadcasting, and for outputting formatted user selected information, and broadcast means for broadcasting the formatted user selected information.

In addition, the present invention also includes an asymmetrical audio program delivery **cellular** system having **mobile** users. The asymmetrical system includes a **mobile** terminal having a receiver receiving user selected program data signals for broadcasting to a **mobile** user, a control processor controlling operations of the **mobile** terminal, and means for receiving a user selection from the **mobile** user, for generating user selection signal inputs responsive to

the user selection, and for transmitting the user selection signal inputs. The **mobile** terminal further includes signal format means for formatting the user selected information for broadcasting, and broadcast means for broadcasting the formatted user selected information. The asymmetrical system also includes a first **mobile** switching office receiving the formatted user selected information broadcast by the broadcast means using a first communication channel which operates under a first communication speed, and information service providers, operatively connected to the first **mobile** switching office. One of the information service providers receives user selection signal inputs received by the first **mobile** switching office from the **mobile** user, and transmits one user selected program responsive to the user selection signal inputs. The asymmetrical system further includes a second **mobile** switching office, operatively connected to at least one of the information service providers, receiving at least one user selected program transmitted by at least one of the information service providers using a second communication channel which operates under a second communication speed. The asymmetrical system is designed so that the second communication speed of the second communications channel of the second **mobile** switching office is substantially greater than the first communication speed of the first communications channel of the second **mobile** switching office.

In another embodiment of the present invention, an audio program and **voice mail** download distribution system having **mobile** users is provided. The download distribution system includes a **mobile** switching office selectively connecting the **mobile** users of the audio program and **voice mail** download system, and information service providers, operatively connected to the **mobile** switching office. One of the information service providers receives user selection signal inputs received by the **mobile** switching office from a **mobile** user, and transmits user selected information to the **mobile** switching office responsive to the user selection signal inputs. The download distribution system also includes at least one **mobile** terminal. The **mobile** terminal includes a receiver, operatively coupling the at least one **mobile** terminal to the **mobile** switching office, receiving the user selected information from the at least one of the information service providers via the **mobile** switching office, and a control processor controlling operations of the at least one **mobile** terminal. The **mobile** terminal also includes means for receiving a user selection from the **mobile** user, for generating user selection signal inputs responsive to the user selection, and for transmitting the user selection signal inputs to the **mobile** switching office, and signal format means for formatting the user selected information for broadcasting, and for outputting formatted user selected information. The **mobile** terminal further includes broadcast means for broadcasting the formatted user selected information, and a memory operatively connected to the receiver. The user selected information comprises at least one of **voice mail** messages and audio programs. In addition, the receiver receives the user selected information corresponding to the user

selection signal inputs and stores the user selected information entirely in the memory before broadcasting to the **mobile** user, thereby minimizing connection between the **mobile** terminal and the **mobile** switching office.

In another embodiment of the present invention, an advanced intelligent network based information distribution system having **mobile** users is provided. The network based distribution system includes a central office switching system connected to communication lines including at least one service switching point for selectively providing switched communications between the communication lines. In addition, the network based distribution system also includes a mobility controller, connected to the central office switching system, arranged for selectively providing **wireless** communications between the central office switching system and **mobile** terminals by using control data conveyed to at least one service switching point through a service transfer point. A network controller is also provided which is arranged for selectively providing control data to effect land line communications, and arranged separately from the central office switching system and the mobility controller. The network controller is connected to both the mobility controller and the service switching point through at least one service transfer point arranged to convey control data to effect communications. The network controller also stores preprogrammed call processing data associated with subscribers who are associated with one of the communication lines connected to the central office switching system and preprogrammed call processing data associated with subscribers who are associated with one of the **mobile** terminals. The network based distribution system further includes information service providers, operatively connected to the mobility controller, at least one of the information service providers receiving user selection signal inputs received by the mobility controller from a **mobile** user, and transmitting user selected information to the mobility controller responsive to the user selection signal inputs. At least one **mobile** terminal is also provided which includes a receiver, operatively coupling the **mobile** terminal to the mobility controller, receiving the user selected information from the at least one of the information service providers via the mobility controller, and a control processor controlling operations of the at least one **mobile** terminal. The **mobile** terminal also includes means for receiving a user selection from the **mobile** user, for generating user selection signal inputs responsive to the user selection, and for transmitting the user selection signal inputs to the mobility controller. Further, the **mobile** terminal includes signal format means for formatting the user selected information for broadcasting, and for outputting formatted user selected information, and broadcast means for broadcasting the formatted user selected information.

In another embodiment of the present invention a method is provided in a radio frequency based information distribution system having **mobile** users. The radio frequency based information distribution system includes a **mobile**

switching office selectively connecting the **mobile** users of the information distribution system, information service providers, operatively connected to the **mobile** switching office, at least one of the information service providers transmitting user selected information to the **mobile** switching office responsive to user selection signal inputs, and at least one **mobile** terminal including a receiver, operatively coupling the at least one **mobile** terminal to the **mobile** switching office, a control processor controlling operations of the at least one **mobile** terminal, means for receiving a user selection from the **mobile** user, signal format means for formatting the user selection for broadcasting, and broadcast means for broadcasting the formatted user selected information. The method of distributing radio frequencies to **mobile** users, includes the steps of:

(a) selectively connecting the **mobile** users of the information distribution system;

(b) receiving a user selection from at least one of the **mobile** users,

(c) generating user selection signal inputs responsive to the user selection,

(d) transmitting the user selection signal inputs to the **mobile** switching office;

(e) receiving user selection signal inputs received by the **mobile** switching office from the at least one of the **mobile** users, and transmitting user selected information to the **mobile** switching office responsive to the user selection signal inputs via at least one of the information service providers;

(f) receiving the user selected information from the at least one of the information service providers via the **mobile** switching office in a receiver of the at least one **mobile** terminal;

(g) controlling operations of the at least one **mobile** terminal;

(h) formatting the user selected information for broadcasting; and

(i) broadcasting the formatted user selected information to the **mobile** user of the at least one **mobile** terminal.

In another embodiment of the invention, a method of distributing communication channels to **mobile** users includes the steps of receiving information signals including at least one of voice, data, and electronic mail signals broadcast from an external source in the receiver of the at least one **mobile** terminal, and storing the information signals in its entirety in at

least one **mobile** terminal prior to broadcasting same to a **mobile** user. The method also includes the steps of controlling operations of the at least one **mobile** terminal, formatting the information signals for broadcasting after being stored in the at least one **mobile** terminal, and broadcasting the formatted information signals to the **mobile** user of the at least one **mobile** terminal.

In another embodiment of the invention, a message download distribution system having **mobile** users is provided. The system includes a **mobile** switching office selectively connecting the **mobile** users of the message download system, and message service providers, operatively connected to the **mobile** switching office. At least one of the message service providers transmits at least one message to the **mobile** switching office. The system also includes at least one **mobile** terminal, operatively coupled to the **mobile** switching office, and receiving the message from the at least one of the message service providers via the **mobile** switching office. The **mobile** terminal formats the message for broadcasting, and broadcasts the formatted message. The **mobile** terminal also includes a memory. The message comprises at least one of voice, data, and electronic mail signals received via at least one of internet, ADSL, another **mobile,** a land-based user, and at least one message service provider for broadcasting to the **mobile** user. The receiver receives the message and stores the message entirely in the memory before broadcasting to the **mobile** user, thereby minimizing connection between said at least one **mobile** terminal and said **mobile** switching office.

In another embodiment of the invention, a method of distributing communication channels to **mobile** users, includes the steps of receiving information signals including at least one of voice, data, and electronic mail signals broadcast from an external source in the receiver of the at least one **mobile** terminal, and storing the information signals in its entirety in at least one **mobile** terminal prior to broadcasting same to a **mobile** user. The method also includes the steps of controlling operations of the at least one **mobile** terminal, and formatting the information signals for broadcasting after being stored in the at least one **mobile** terminal. The method also includes the steps of receiving supplemental information including destination information and optionally additional information from the **mobile** user, and associating the formatted information with the supplemental information, the destination information including at least one of internet address, an ADSL address, another **mobile,** a land-based user, and a message service provider. The method also includes the step of at least one of broadcasting the formatted selected information to the **mobile** user, and broadcasting the formatted and supplemental information to the at least one of the internet address, the ADSL address, the another **mobile,** the land-based user, and the message service provider.

In another embodiment of the invention, a method of distributing

communication channels to **mobile** users includes transmitting information signals including at least one of Web related information and Internet related information received via one ore more of the Internet, ADSL, another **mobile,** a land-based user, and the at least one information service provider. The method also includes receiving the information signals from one or more of the Internet, ADSL, another **mobile,** a land-based user, and the information service provider in the receiver of the **mobile** terminal, and storing the information signals in its entirety in the **mobile** terminal prior to broadcasting and/or displaying same to the **mobile** user. The method also formats, broadcasts and/or displays the information signals after being stored in the **mobile** terminal.

These, together with other objects and advantages which will be subsequently apparent, reside in the details of construction and operation as more fully hereinafter described and claimed, with reference being had to the accompanying drawings forming a part hereof, wherein like numerals refer to like elements throughout.

DRAWING DESCRIPTION:
BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a conceptual diagram of the **mobile** audio program selection system of the present invention;

FIGS. 1A-1B are flowcharts of the **cellular** communication process used in conjunction with MAPOD of the present invention;

FIG. 2 is a block diagram of the **Mobile** Audio PrOgram selection Device (MAPOD) of the present invention;

FIG. 3 is a block diagram of the **mobile** audio program provider of the present invention;

FIG. 4 is a block diagram of another embodiment of the **mobile** audio program selection device of the present invention;

FIG. 5 is a block diagram of another embodiment of the **mobile** audio program selection device of the present invention;

FIG. 6 is a block diagram of another embodiment of the **mobile** audio program selection device of the present invention;

FIG. 7 is a block diagram of another embodiment of the **mobile** audio program selection device of the present invention;

FIG. 8 is a diagram of another embodiment of the **mobile** audio program

selection system of the present invention;

FIG. 9 is a diagram of another embodiment of the **mobile** audio program selection system of the present invention;

FIGS. 10A-10B together comprise a block diagram of the circuit construction of the present invention according to the embodiment of FIG. 9;

FIG. 11 is a flowchart describing the **mobile** access attempt process of the present invention;

FIG. 12 is a flowchart describing the **mobile** registration message process of the present invention;

FIG. 13 is a flowchart describing the **mobile** origination message process of the present invention;

FIG. 14 is a flowchart of the **mobile** page response message process of the present invention;

FIG. 15 is a flowchart of a route attempt process of the present invention;

FIG. 16 is a flowchart of the **mobile** de-registration process of the present invention;

FIG. 17 is a block diagram of a standard ADSL arrangement;

FIG. 18 is a diagram of Internet 2 architecture;

FIG. 19 is an illustration of the architecture of the combined internet, internet 2, POTS, and ADSL architecture; and

FIGS. 20-23 are flowcharts describing in detail the process flow of the **mobile** telephone receiving an incoming message comprising one or more of a voice message, a data message, an electronic mail message, an internet message, and/or and ADSL message for storage on the handset, and also for subsequent uploading to another destination;

FIG. 24 is an illustration of the software implementation architecture for the Java language;

FIG. 25 is an illustration of an open architecture software implementation architecture for the Java language using the standard Active X protocol approach;

FIG. 26 is an illustration of conceptual framework for the virtual reality modeling language (VRML);

FIG. 27 illustrates a sample application using a browser as a front-end to a database;

FIG. 28 shows how the Java JDBC Application Programmer Interface (API) allows the time card application to communicate directly with the DBMS, eliminating many of the problems of a traditional Web-based system;

FIG. 29 is an illustration of the software execution of a Java applet for the Java language; and

FIGS. 30-34 are flowcharts describing in detail the process flow of the **mobile** telephone interactively receiving information comprising one or more of data, a voice message, a data message, an electronic mail message, an internet message, and/or and ADSL message for storage on the handset of the **wireless** device, and also for optional subsequent uploading to another destination.

DETAILED DESCRIPTION:

DESCRIPTION OF THE PREFERRED EMBODIMENTS

FIG. 1 is a conceptual diagram of the **mobile** audio program selection system of the present invention. In FIG. 1, a **Mobile** Audio Programming Device (MAPOD) 2 is located in close proximity to a user, e.g., in an automobile or strapped around a user's waist. MAPOD 2 will interface with standard **cellular** network 4 or an optional asymmetrical/one-way downstream network 20.

The standard protocol which MAPOD 2 uses to interface with standard **cellular** network 4 or asymmetrical/one-way downstream network 20 is preferably the standard protocol used by **cellular** telephones. This standard protocol is described in detail, for example, in EIA/TIA publications IS-41.1-A, IS-41.2-A, IS-41.3-A, IS-41.4-A, and IS-41.5-A. Other interface protocol and systems are presented in U.S. Pat. Nos.: 5,371,898; 5,369,681; 5,353,352; 5,353,331; 5,307,400; 5,257,400; 5,251,249; 5,247,698; 5,119,502; 5,119,397; 5,111,534; 5,020,093; 5,020,092; 5,020,091; 5,008,925; 4,972,455; 4,905,301; 4,893,327; 4,799,253; 4,754,495; 4,750,198; 4,599,490, all incorporated herein by reference. A brief summary is presented here as the protocol pertains to the interface between MAPOD 2 and **cellular** networks 4 and 20.

A typical **cellular mobile** radio telephone system is controlled by at least one **mobile** switching center (also known as a **mobile** telephone switching office), at least one base station, and at least one **mobile** station. The **mobile** switching center constitutes an interface between the radio system and the public switching telephone network. The base station transmits information

between the **mobile** stations and the **mobile** switching centers. Calls to and from **mobile** subscribers are switched by the **mobile** switching center.

The **mobile** switching center also provides all signalling functions needed to establish the calls. In order to obtain radio coverage of a geographical area, a number of base stations are normally required. This number may range from, in the exceptional case, one base station, up to one hundred or more base stations in normal systems. The area is divided into cells, where each cell may either be serviced by a base station or may share a base station with a number of other cells.

Currently, **cellular** radiotelephone service is provided in the 825 to 845 Mhz and 870 to 890 Mhz frequency bands. The higher-frequency band is used for "down-link" transmissions from the "cell site" for reception by the subscriber. The cell site is the location of the transmitter, or, more specifically, the location of the antenna from which transmissions are effected for the cell. The lower frequency band is used for "up-link" transmissions from the subscriber in the cell for reception by the receiving equipment which is also located at the cell site.

Each frequency band assigned to the **cellular** radiotelephone system is divided into two groups, with one group being reserved for the local telephone company and the other group being franchised to a completing service provider. Each **cellular** channel has a thirty kilohertz bandwidth, allowing for 666 sequentially numbered channels, with channels 1 through 333 being allocated to one service provider and channels 334 through 666 being allocated to the other service provider.

Communication between the radio base stations within the system and the **mobile** stations within the system are divided into a plurality of voice or speech channels and at least one access or control channel, which may be either analog or digital and which may have any data rate. An illustrative one of such access or control channel is referred to as the forward control channel (FOCC).

Each **mobile** station which is operating within a **cellular** communications system must be locatable when a call is received by the system which is intended for that station. A **mobile** station is located by broadcasting a paging signal directed to the **mobile** and requesting it to respond if it receives the page. When the **mobile** broadcasts its page response signal to the page signal it is then placed on a voice channel by the base station and the call intended for the **mobile** can be connected to it through that voice channel. **Cellular** telecommunications systems employ a control channel such as the forward control channel (FOCC) as the means by which paging signals are broadcast into the various cells of the system in order to locate a particular

mobile station.

The control channel, such as the FOCC, is typically restricted to a rate on the order of 8-10 K bits per second which is a speed limitation imposed by the technology used in that implementation. The control channel may also be utilized to transmit other messages to the **mobile** stations, including, for example, voice channel designations, directed retry orders, system ordered rescan signals and system overhead message trains each of which use substantial control channel capacity each time they are transmitted.

Paging provides the ability to locate a **mobile** station's whereabouts within the exchange in order to set up a call to that **mobile** station. More specifically, the paging process in **mobile cellular** radio systems, attempts to identify the specific cell containing that **mobile,** as described above in connection with the paging process. During the execution of this process, the **mobile** switching center (MSC) searches for the **mobile** by sending a sequence of paging messages on the FOCC of the system and awaits a page response. Obviously, the page message muse be transmitted to all of the cell sites covering the entire service area of the system in order to ensure that the **mobile** is located regardless of where it might be within the system.

In present systems, when a page remains unanswered by the **mobile** station which is sought, the page must be repeated. This repetition can be either within a location area previously paged or within the entire service area (SA) of the system. The present practice within **cellular** radio systems is to employ the paging process to handle incoming page requests on a "first come, first served" basis. Depending upon whether the location area (LA) of the requested **mobile** station is known or not, the amount of paging capacity allocated to serve a particular page request is the same. That is, if the LA of the **mobile** station is known, then the first page attempt is within the LA. Otherwise, the page attempt is within the service area (SA) which includes all of the LA's within the exchange. If no response is received to the page, the page is repeated either within the LA itself or within the SA.

When attempting to route a call to a **mobile** station, the MSC must specifically know in which cell the **mobile** station is located. In accomplishing the task of locating the **mobile,** the MSC pages the **mobile** station in the location area where the **mobile** station last registered. This prevents a global or system-wide page wherein all the cells within an exchange are paged simultaneously. If the **mobile** station does not answer the page request in the registered location area of its last registration, only then is service area or global paging required in order to locate the **mobile**.

A known solution to the problem of locating the **mobile** phone is based on the concept of **mobile** registration. **Mobile** registration is the process by which a

mobile phone becomes listed as being present in the service area of one of the **mobile** exchanges in a **mobile** telephone service network. It should be recognized that one purpose of **mobile** registration is to permit calls to a **mobile** phone to be automatically delivered even though the **mobile** phone may be moving from place to place through a network of **cellular** systems.

It should also be recognized that **mobile** phone registration according to EIA Standard IS-3D is effected by means of interactions between the **cellular** system and the **mobile** phones operating in its service area. One such interaction is called "autonomous registration" and it is controlled by the **cellular** system through certain information transmitted to the **mobile** phones. This information is in the form of an overhead message train (OMT), which is transmitted on paging channels throughout a **cellular** system service area, normally once each second approximately. The OMT includes a system parameter overhead message including station and registration related messages, and optionally, several other messages of which the registration identification message and the registration increment message relate to the autonomous registration process.

Registration may be enabled or disabled individually for each class of **mobile** phone, e.g., home or roam (explained below), by means of control bits in the system parameter overhead message. The system parameter overhead message also contains the identification number of the serving **cellular** system from which the **mobile** phone determines whether it is a "home" or a "roam" **mobile** phone. Each **mobile** phone contains, in its internal memory, an entry indicating the identity of its home **cellular** system and an entry indicating the **cellular** systems (which may be the home **cellular** system) in which it has most recently registered successfully. It also stores a value for the **cellular** system used to determine when it is scheduled to re-register in that **cellular** system.

In the **mobile** telephone systems used in North America, the United Kingdom and in other markets, twenty-one frequencies are allocated for the control channels. A two-bit digital color code (DCC) is used to differentiate control channels using the same frequency. It is thus possible to have up to 84 cells, each cell having a control channel with a unique set of frequency and DCC combinations. In densely populated areas, subscriber demand may require more than 84 cells to provide adequate **mobile** telephone service.

In FIG. 1, standard **cellular** network 4 interfaces with MAPOD 2, preferably as described above. In particular, **cellular** antenna 6 receives signals from MAPOD 2, and transmits signals generated by **mobile** switching office 8. **Mobile** switching office 8 permits connection between MAPOD 2 and application provider 12 via tandem switching office 10. Application provider 12 permits and facilitates selection of various audio data by the user of MAPOD 2 via speech recognition/synthesized audio response units 14a, 14b which then convert audio commands issued by the user of MAPOD 2 into an audio request. One example of

an intelligent voice recognition system is described in commonly assigned application Ser. No. 08/271,887, filed Jul. 7, 1994, entitled "Intelligent Recognition" (attorney docket no. 680-091A), the disclosure of which is incorporated herein entirely by reference. Application gateway 16 coordinates and supervises the audio data requests issued by MAPOD 2 and transmits the audio request to programming provider 18. Programming provider 18 includes programming server 20 which contains the audio data requested by MAPOD 2 and which transmits the audio data back to application provider 12.

Application provider 12 then transmits the audio data to either standard **cellular** network 4 or to optional asymmetrical **cellular** network 20 which may be dedicated for transmitting the audio data from application provider 12 to MAPOD 2. Asymmetrical **cellular** network 20 includes tandem switching office 22 which receives the audio data from application provider 12 and transmits the audio data to MAPOD switching office 24. MAPOD switching office 24 will then transmit the audio data via **cellular** antenna 26 to MAPOD 2. Since MAPOD switching office 24 is dedicated to transmission of audio data, MAPOD switching office 24 preferably utilizes a higher bit rate channel, for example, having a larger band width capacity, to transmit the audio data to MAPOD 2. Accordingly, the **mobile** audio program selection system of the present invention utilizes the public switch telephone network (PSTN) in order to receive the audio request by the user of MAPOD 2, and transmits the audio data to MAPOD 2 via either standard **cellular** network 4 or asymmetrical **cellular** network 20.

FIGS. 1A-1B are flowcharts of the **cellular** communication process used in conjunction with MAPOD of the present invention. In FIGS. 1A-1B, each **mobile** station which is operating within a **cellular** communications system must be locatable when a call is received by the system which is intended for that station. A **mobile** station is located by broadcasting a paging signal directed to the **mobile** in step R2. The **mobile** is requested to respond if it receives the page in step R4. The page message is transmitted to all of the cell sites covering the entire service area of the system in order to ensure that the **mobile** is located regardless of where it might be within the system in step R6.

When the **mobile** broadcasts its page response signal to the page signal it is then located in step R8. In addition, the specific cell where the **mobile** is located is also identified in step R10. The **mobile** is placed on a voice channel by the base station in step R12, and the call intended for the **mobile** can be connected to it through that voice channel in step R14. **Cellular** telecommunications systems employ a control channel such as the forward control channel (FOCC) as the means by which paging signals are broadcast into the various cells of the system in order to locate a particular **mobile** station.

Paging provides the ability to locate a **mobile** station's whereabouts within the exchange in order to set up a call to that **mobile** station. More

specifically, the paging process in **mobile cellular** radio systems, attempts to identify the specific cell containing that **mobile,** as described above in connection with the paging process. During the execution of this process, the **mobile** switching center (MSC) searches for the **mobile** by sending a sequence of paging messages on the FOCC of the system and awaits a page response.

When a page remains unanswered by the **mobile** station which is sought in step R16, the page must be repeated. This repetition can be either within a location area previously paged or within the entire service area (SA) of the system. Depending upon whether the location area (LA) of the requested **mobile** station is known or not, the amount of paging capacity allocated to serve a particular page request is the same. That is, if the LA of the **mobile** station is known, then the first page attempt is within the LA. Otherwise, the page attempt is within the service area (SA) which includes all of the LA's within the exchange in step R18. If no response is received to the page, the page is repeated either within the LA itself or within the SA.

Mobile registration is the process by which a **mobile** phone becomes listed as being present in the service area of one of the **mobile** exchanges in a **mobile** telephone service network. **Mobile** phone registration according to EIA Standard IS-3D is effected by means of interactions between the **cellular** system and the **mobile** phones operating in its service area. One such interaction is called "autonomous registration" and it is controlled by the **cellular** system through certain information transmitted to the **mobile** phones in step R20. This information is in the form of an overhead message train (OMT), which is transmitted on paging channels throughout a **cellular** system service area, normally once each second approximately.

The OMT includes a system parameter overhead message including station and registration related messages, and optionally, several other messages of which the registration identification message and the registration increment message relate to the autonomous registration process in step R22.

Registration may be enabled or disabled individually for each class of **mobile** phone, e.g., home or roam (explained below), by means of control bits in the system parameter overhead message in step R24. The system parameter overhead message also contains the identification number of the serving **cellular** system from which the **mobile** phone determines whether it is a "home" or a "roam" **mobile** phone in step R26. Each **mobile** phone contains, in its internal memory, an entry indicating the identity of its home **cellular** system and an entry indicating the **cellular** systems (which may be the home **cellular** system) in which it has most recently registered successfully. It also stores a value for the **cellular** system used to determine when it is scheduled to re-register in that **cellular** system.

FIG. 2 is a block diagram of the **mobile** audio program selection device. In FIG. 2, MAPOD 2 receives and transmits signals via **cellular** antenna 22. Standard **cellular** interface 24, as described previously, will transmit the audio request via, for example, handset 42 or hands-free microphone 44. In addition, **cellular** interface 24 receives audio data via **cellular** antenna 22. Controller 26 coordinates, monitors and controls the broadcasting of audio data received or transmitted via **cellular** interface 24. **Cellular** interface 24 then transmits the audio data to MAPOD digital interface 28 which demodulates the data in a standard form for broadcasting to the user. MAPOD digital interface 28 may be, for example, a standard modem which demodulates the received data.

MAPOD digital interface then transmits the demodulated data to decoder 30 which decodes the encoded data in accordance with a predetermined coding scheme. For example, suitable video coding algorithms rely on motion compensated prediction (MCP) and motion compensated interpolation (MCI). Motion compensated predictive/interpolative coding (MCPIC) is described in Wonq et al. "NCPIC: A video coding algorithm for transmission and storing applications", November, 1990 IEEE Communications magazine. While the above coding algorithms relate to video data, they may be also utilized with respect to audio data.

Another **compression** technique using motion estimation, motion compensation predictive coding and adaptive discrete cosine transform quantization is supported by the International Standards Organization (ISO) moving pictures expert group (MPEG). MPEG-1 specifies a coding algorithm having a data rate of 1.2 MBPS. This digital impression standard may be accommodated by a data channel having the capability of 1.544 MBPS. MPEG programmable decoder/processors, capable of decompressing digital data in real time, have been produced by such companies as C-Cube Microsystems and LSI of San Jose, Calif.

Decoder 30 outputs the audio data via standard preamp and RF output conductors 32 to MAPOD switch 34. MAPOD switch 34 which is controlled by controller 26 then outputs the audio data to amplifiers and/or speakers for example, in an automobile or to auxiliary speakers attached to MAPOD 2. Advantageously, MAPOD switch 34 includes the capability of receiving radio frequencies (RF) from a local antenna, and the capability of receiving local preamplified signals as well. Thus, for example, MAPOD switch 34 may be used to either transmit audio data received from a program provider via a **cellular** network, or audio data received via, for example, a local radio in an automobile or positioned locally with respect to MAPOD 2. MAPOD 2 also includes control interface 40 for programming or specifying specific instructions for controller 26. For example, controller 26 may be programmed to play the audio data received via a **cellular** network even when audio data is also received simultaneously via a local radio by MAPOD switch 34. In this

situation, controller 26 considers the audio data received from the **cellular** network of a higher priority.

FIG. 3 is a block diagram of the **mobile** audio program provider system. In FIG. 3, transmitter/receiver (transceiver) 46 is designed to communicate with the transceiver, i.e., standard **cellular** interface 24, in MAPOD 2. Transmitter/receiver 46 provides connection with the public switch telecommunications network (PSTN) as will be described. Transmitter/receiver 46 is adapted to transmit **compressed** data to the transceiver in the MAPOD, and to receive **compressed** digital data transmitted by the MAPOD.

The **compressed** data received via a receiver section of transmitter/receiver 46 is fed to buffer storage 45 under the control of processor controller 48. The stored signal is then fed to encoder/decoder 50 which decodes the received signal, if needed, into a format which is acceptable for application provider network 52. Application provider network 52 includes switching functions for switching between application providers 12a, 12b. Application provider network 52 preferably includes a conventional message system platform including voice processing functions and storage. Depending upon the address attached by application provider network 52, the audio request may be stored for automatic delivery using conventional call completion services. Once delivered to the appropriate application provider 12a, 12b, an appropriate program server, for example, 20a, 20b is accessed for the requested audio selection. The audio data is then transmitted from the selected program server to the application provider network 52 via the appropriate application provider, and is encoded for transmission to the **mobile** audio program device 2.

FIG. 4 is a block diagram of another embodiment of the **mobile** audio program device which includes standard **cellular** equipment for voice connections to other parties. In FIG. 4, a portable transceiver device is illustrated which may be installed in an automobile vehicle in a similar manner as a **cellular** telephone or which may be battery powered and completely portable, as is also common with standard **cellular** telephones. The portable transceiver device includes hand set 56 which includes a conventional input or a microphone 60, and an output or ear piece 62. Actuation button 58 is provided for a purpose to be described below.

The output of hand set 56 comprises an analog voice signal which is fed to standard **cellular** equipment 70 via switch 66 under the control of controller 68. The analog voice signal will typically be fed to **cellular** equipment 70 when controller 68 determines that MAPOD 2a is not being used and a standard **cellular** telephone call is being performed or when hand set 56 transmits digit data indicating a telephone call is to be initiated by the user. **Cellular** equipment 70 will then format the analog voice signal or digit data for transmission by transmitter 76 and antenna 78 via switch 72 which is also

controlled by controller 68. In this manner, standard **cellular** telephone calls or telephone calls which are initiated by the user of hand set 56 will be executed by **cellular** equipment 70.

Receiver 80 will receive a **compressed** signal either for a standard **cellular** telephone connection from another party or will receive audio data from an application provider via an application provider network. Switch 72 directs the received information either to standard **cellular** equipment for standard **cellular** telephone call or to MAPOD 2a for transmitting of audio data received from an application provider to the MAPOD user. Switch 72 is under the control of controller 68 and may be preset in accordance with, for example, switch activation button 58 of hand set 56, or may be switched in accordance with the content of the signal received by receiver 80 and monitored by controller 68.

When controller 68 determines that the received information is audio data to be formatted and presented to the user of hand set 56, the received information is transmitted to storage 82 and accessed by decoder/encoder 84 for decoding the received **compressed** data. The decoded data is then transmitted to A/D converter 86 which will convert the decoded data from digital to analog form and transmit the analog converted audio data to either output 62 of hand set 56 or to input/output (I/O) switch 88 which under the control of controller 68. Switch 88 is used to transmit audio data to broadcast devices via output 87, as well as accept radio frequencies from a local antenna at input 89 as discussed previously. When the audio data received from the application provider is a menu listing the various audio selections available to the user, the user may input a selection either by voice via input 60 of hand set 56, or by transmitting digits input via handset 56. This selection information is then transmitted back to the application provider, and based upon the selection, the application provider will arrange to obtain and provide the selected audio data from a program server to the user of MAPOD 2a.

MAPOD 2a advantageously includes the capability of receiving large amounts of data which are downloaded from the application provider for the selected audio program. Specifically, MAPOD 2a may either transmit the audio data received from the application provider in real-time to the MAPOD user, or MAPOD 2a may receive all the audio data via a high data rate channel and store the **compressed** information in storage 82. The **compressed** audio data may then be retrieved and decoded by decoder/encoder 84, and transmitted to the MAPOD user at a pre-selected time. Once the audio program is downloaded to MAPOD 2a and stored in storage 82, MAPOD 2a includes the additional feature of severing the connection between MAPOD 2a and the application provider in order to minimize the amount of **cellular** connection, and the amount of cost which is charged to the user of MAPOD 2a.

Additionally, MAPOD 2a may be located within, for example, a stereo system

or radio which will receive the audio data from the application provider directly, and broadcast the audio data on associated speakers with the stereo system. In this instance, hand set 56 would be used to order or select the audio data from the application provider via standard **cellular** equipment, and the application provider would communicate directly with MAPOD 2a once the selection has been made for the broadcasting of the audio data. Similarly, MAPOD 2a may be used in conjunction with a **voice mail** system where the user of MAPOD 2a will have their voice messages downloaded from a **voice mail** system using a high data rate channel, and which are later broadcasted for the user. According to this scenario, the user of MAPOD 2a will minimize the **cellular** connection with the **voice mail** system and may have their messages broadcasted at a later time, or repeatedly without incurring substantial **cellular** telephone connection charges which would otherwise be experienced.

FIG. 5 is a block diagram of another embodiment of the **mobile** audio program selection device of the present invention. In FIG. 5, the elements designated by the reference numerals which are identical to FIG. 2 perform the similar function, and therefore, will not be discussed in detail. In FIG. 5, MAPOD unit 2b includes, instead of a standard **cellular** interface 24, a two line standard **cellular** interface 90 which permits simultaneous voice connection between the MAPOD user and another party as well as an additional line which would permit the connection between a MAPOD user and the application provider network or another party. Two line **cellular** interface 90 is standard and will not be discussed here in detail.

FIG. 6 is a block diagram of another embodiment of the **mobile** audio program selection device. In FIG. 6, MAPOD unit 2c is substantially the same as MAPOD 2a illustrated in FIG. 2. However, MAPOD 2c further includes an asymmetrical one-way **cellular** interface 92 which is used for specifically receiving asymmetrical one way **cellular** audio data from an asymmetrical **cellular** network which is dedicated for the transmission of the audio data. In this situation, asymmetrical **cellular** interface 92 is able to receive the audio data at a much higher speed thereby further minimizing the connection time between MAPOD 2c and the **cellular** network which is transmitting the audio data. U.S. Pat. No. 5,247,347 is one example of an asymmetric connection between subscriber and application provider using standard asymmetrical digital subscriber line interface units over local lines, which is incorporated herein by reference.

In the previously described environment in connection with FIGS. 2 and 4-6, the MAPOD User Terminal would generally have the following capabilities:

(1) **Cellular** interface terminal

(1a) standard (FIGS. 2 and 4), or

(1b) two-line standard (FIG. 5), or

(1c) standard plus asymmetrical/one-way downstream (FIG. 6)

(2) MAPOD digital interface to standard or optional one-way/asymmetrical **cellular**

(3) **Compressed** audio decoder

(4) Local audio pass-through

These capabilities are available to the user within the following operating modes:

(a) Telephone call only: (1a), (1b), or (1c)

(b) MAPOD only: (1a), (1b), or (c); and (2)-(3)

(c) Telephone call and MAPOD: (1b) or (1c); and (2)-(3)

(d) Telephone call and local audio: (1a), (1b), or (1c) and (4)

(e) Off: (4)

The simplest situation involves a user terminal with (1a) type capability. The MAPOD application takes place over a standard **cellular** call. Operating modes (a), (b), (d) and (e) are possible. To access a MAPOD application, the user places a standard **cellular** call to the information provider and then negotiates service with the Application Gateway via the Speech Recognition Synthesized Audio Response (SR/SAt) unit. once an application is selected, it is provided over the same call path.

With a type (1b) user terminal, operation is similar to the above except that operating mode (c) is now available due to the presence of a second line.

With a user terminal possessing type (1c) asymmetrical or one-way downstream capability, the application negotiation phase would take place over a standard **cellular** call. For low- or non-interactive applications, application delivery would take place over a one-way or asymmetrical **cellular** call (established by the Application Gateway), thus freeing up the standard **cellular** interface. All operating modes are possible.

A typical MAPOD session could operate as follows from the user's perspective:

1. A user is driving an automobile and has activated the MAPOD device.

2. The MAPOD device initiates a **cellular** call to the MAPOD server gateway Speech Recognition of Synthesized Audio Response (SR/SAR) unit.

3. The SR/SAR welcomes the user and provides an audio menu (e.g., music, library, etc.). This menu could be standard or pre-customized by the user. Also, the experienced user could interrupt the menu to move more quickly through the process.

4. The user makes a selection by speaking the desired item.

5. The SR/SAR continues to prompt the user with audio menus and interpreting the verbal response until a specific programming selection is made (e.g., Glen Miller medley). The gateway application then prompts the server to provide the requested programming.

6. The user may interpret programming delivery at any time with the appropriate verbal command or by terminating the session (i.e., hanging up). In order to prevent an inadvertent interruption, the user could activate a switch on the MAPOD device to temporarily disable voice command transmission.)

Since the MAPOD delivery system is not limited by economics to programming formats with mass appeal as with broadcast radio, there is the potential to satisfy the needs of many more users who may desire programming alternatives. The proposed system could provide individual users with access to large programming libraries.

FIG. 7 is a block diagram of another embodiment of the **mobile** audio program selection device having the ability to present a limited amount of graphical information to the user. In FIG. 7, MAPOD 2d may connect or interface with a number of different types of application provider networks, such as described in commonly assigned application Ser. No. 08/250,792, filed May 27, 1994, entitled "Full Service Network" (attorney docket no. 680-080), and commonly assigned application Ser. No. 08/250,791, filed May 27, 1994, entitled "Dynamically Programmable Digital Entertainment Terminal" (attorney docket no. 680-083), the disclosures of which are incorporated herein entirely by reference.

For each different type of network, MAPOD 2d includes transceiver 100 providing the actual physical connection to the particular type of network. Transceiver 100 will also perform any format conversion necessary between signal formats utilized by the network and signal formats used within MAPOD 2d. Transceiver 100 also provides two-way signal conversion and formatting, for example, for a control signalling channel and other standard **cellular** protocol

described previously.

In the illustrated embodiment, transceiver 100 presents two connections to the rest of MAPOD 2d, a high bit rate broadband connection and a low bit rate signaling connection. The broadband connection is a one-way downstream only connection, but the low-bit rate signaling connection is a two-way connection.

Transceiver 100 may take the form of a plug in module. In the preferred embodiment, transceiver 100 would be similar to a daughter board or option card which can be plugged into a back plane of a personal computer (PC). In such an embodiment, typically a technician could replace the module in either the field or the shop, to modify transceiver 100 to connect to and communicate over a different network, and the technician would modify associated communications control software in the system memory. Alternative implementations may use a user replaceable cartridge type network interface module, similar to a video game cartridge, which may include memory in the module for storage of the communications control. As a further alternative, the network interface module could include a digital signal processor controlled by the CPU of the transceiver 100, and input/output connections compatible with all of the digital broadband networks currently available. The downloaded operating system software stored in the system memory of the transceiver would control operations of the digital signal processor to send and receive signals in accord with the particular network the subscriber chooses to connect with transceiver 100.

MAPOD 2d includes CPU 98, comprising, for example, a 386 or 486 microprocessor 104 and associated system memory 106. The system memory 106 preferably includes at least 2 Mbytes of volatile dynamic RAM 110 and 1 Mbyte of non-volatile RAM 108. The microprocessor 104 also includes a small amount of ROM (not shown) storing "loader" programming needed to control "wake-up" after the power is turned "on". An EPROM memory (not shown) also may be added.

A digital audio/picture signal processor 96, controlled by the CPU 98, produces digital uncompressed audio and picture or graphical signals from the audio and picture MPEG encoded packets received from the network through transceiver 100. The audio/picture processor 96 includes an MPEG system demultiplexer 102, an MPEG picture decoder 112, an MPEG audio decoder 114, a graphics overlay controller 118 and at least two frames (e.g. 8 Mbytes) of picture RAM 116.

The MPEG system demultiplexer circuitry 102 recognizes packets in the MPEG data stream received over the broadband channel through transceiver 100, and routes the packets to the appropriate components of MAPOD 2d. For example, the MPEG system demultiplexer 102 circuitry recognizes audio and picture packets in the MPEG data stream and routes those packets to the decoders 114 and 112,

respectively.

The MPEG picture decoder 112 decompresses received picture or graphical packet signals to produce a digital signal, and the MPEG audio decoder 114 decompresses received audio packets to produce left and right digitized stereo signals. For at least some functions, the MPEG decoders 112, 114 may be controlled in response to signals from the microprocessor 104. The MPEG picture decoder 112 will internally include at least two frames (e.g. 8 Mbytes) of RAM (not separately shown) for use as a frame reorder buffer during the MPEG decoding process, and the MPEG audio decoder 114 also may include some buffer memory.

The picture RAM 135 is preferably a standard digital data RAM, of appropriate size, which is used in MAPOD 2d to store digitized frames of video data. The RAM within the MPEG picture decoder 112 likewise consists of standard digital data RAM.

The graphics overlay controller 118 produces displays of text and graphics data, such as the initial turn-on selection menu received over the signaling channel, in response to instructions from the CPU 98. The picture RAM 116 sequentially receives each frame of digitized, uncompressed video information, as output from the MPEG picture decoder 112. The picture RAM 116 also receives digital information and read/write control signals from the graphics overlay controller 118 representing the several planes of text and graphics information and combines that information with the frames of decompressed picture to produce composite picture frames.

The graphics overlay controller 118 and the picture RAM 116 cooperate to manipulate, for example, five different planes of video information, four of which may be active at any one time, to produce the composite picture frame output signals. The individual planes comprise the decoded MPEG picture frames, a cursor, two graphics/text image planes manipulated by the microprocessor 104 and a backdrop plane. The backdrop plane would be switched in to replace the plane representing the decoded MPEG picture frames, e.g. to present a blue background instead of the MPEG picture background.

When there are no graphics or text, the composite frames would correspond entirely to the uncompressed received picture frames output by the MPEG picture decoder 112. When no received picture frames are to be output, either when none are received or when they are to be entirely replaced, the information from the graphics overlay controller 118 specifies a background and the active planes of text or graphic information. When received picture frames are combined with text and/or graphics, the composite picture frames include the uncompressed received picture frames with selected pixels thereof replaced with graphics or textual data display pixels specified by the graphics overlay

controller 118. In this last situation, the graphics overlay controller 118 would deactivate the backdrop plane.

MAPOD 2d also includes audio and picture digital to analog converters and appropriate drivers to produce output signals compatible with a conventional television set or monitor. Specifically, the converter and driver circuitry of MAPOD 2d includes audio digital to analog converters (DAC) 126, 128, an audio mixer 130, an NTSC encoder 120, and an RF (radio frequency) demodulator 122.

The DAC's 126 and 128 receive the uncompressed left and right digitized audio signals output by the MPEG audio decoder 114. In response, the DAC's 126 and 128 produce baseband analog audio signals for output to individual baseband output terminals. The audio mixer 130 also receives the baseband audio signals from the DAC's 126 and 128. The mixer 130 combines the left and right analog audio signals to produce a monaural audio signal as the audio input to demodulator 122 which is synchronized via RF oscillator 124.

The NTSC encoder 120 also performs a digital to analog converter (DAC) function. In response to the digitized picture signals received from the picture RAM 116, the NTSC encoder 120 produces a baseband analog signal in standard NTSC format. The baseband NTSC signal is supplied to an output terminal 132 of MAPOD 2d. The baseband NTSC video signal is also supplied to the RF demodulator 122. The RF demodulator 122 responds to the mono audio signal, the NTSC signal and an RF signal from a local RF oscillator 124, to produce a standard RF television signal on an available TV channel, typically channel 3 or channel 4.

The type of connection of MAPOD 2d to the television set or monitor depends on the capabilities of the user's television set. If the user has a monitor type television capable of receiving baseband picture and stereo audio inputs, the appropriate terminals of the television would connect directly to the picture and audio output terminals 132 and 134 of MAPOD 2d. If the subscriber does not have such a television monitor, then the RF output of the demodulator 122 would be connected to the cable or antenna input connection of the television, e.g. by coaxial cable via RF output 136. Alternatively, the digitized picture and audio may go to separate output terminals (not shown) for connection to inputs of digital display devices, for example, for high definition television (HDTV) sets.

MAPOD 2d is an open interface device in that it interacts with equipment of a large number of program providers to offer users a wide array of principally audio programming for the **mobile** user. MAPOD 2d is preferably a programmable device to which different individual program providers can download application software, and at least one program provider can download all or a part of the operating system. In non-volatile memory (ROM and non-volatile RAM), MAPOD 2d

will store a loader program and an operating system. The loader program and operating system in the ROM and the non-volatile RAM will include sufficient programming to control initial communications and define interfaces and drivers.

MAPOD 2d also includes a magnetic card reader 135 connected to the microprocessor 104. This reader 135 could be used to scan credit card information encoded on magnetic strips on commonly available credit cards for purchasing audio programming. In a home shopping and purchasing audio service, controlled by the downloaded software, the user would scan their own credit card through the magnetic card reader 135 as part of the payment operations. The reader could also have magnetic write capabilities to perform debit card operations.

MAPOD 2d further includes a personal computer memory-card interface adapter (PCMCIA) port 137. This is a two-way interface for connection to and communication with a flash memory module, such as is now incorporated into advanced "smart card" devices. A user might communicate with an auxiliary database connected via PCMCIA port 137 and a broadband network. For example, the user's personal information could be read from the smart card and subsequently updated on the smart card, through the PCMCIA port 137. Another use of this port might involve communication to another system to download information. Although specified as a "memory" port and mapped by the CPU as part of its system memory space, the devices connected to this port 137 can have other data processing capabilities, e.g. buffering and modem communication capability.

In the current implementation, the PCMCIA port 137 will carry 6 Mbits/s of data, but the port can be designed for higher speeds such as 20 Mbytes/s. Another use of this port would be for connection to an Ethernet card or other Local Area Network (LAN) card to permit data communications between MAPOD 2d and one or more computers. MAPOD 2d would provide the computers with communication services through the broadband network, for example to receive high speed downloads of new or updated software for those computers.

FIG. 8 is a diagram of another embodiment of the **mobile** audio program selection system used in connection with an Advanced Intelligent Network (AIN) type architecture. In FIG. 8, one or more central office switches, such as the class 4/5 Switch 160, are located throughout a state or region served by a telephone operating company (TELCO). Local telephone lines connect the central office switch 160 to individual telephone terminals in each geographic area, for example to the Plain Old Telephone Service (POTS) phone 166.

Although shown as telephones in FIG. 8, the terminals can comprise any communication device compatible with the line. In addition, **wireless**

communication services are provided via radio links using frequencies assigned to **cellular** communications networks. Other types of **wireless** communication, however, could be substituted for the radio communication systems. For example, the invention could use a series of radio relay transponders, an infrared system or a satellite based system to provide one or more of the **wireless** links.

Switch 160 connects via trunk circuits 158, 176 to one or more Mobility Controllers (MC's), such as the **Cellular** MC 138 and the Personal Communication Service (PCS) MC 170. Each central office my also connect via trunk circuits to one or more remote central offices. The trunk circuits carry large numbers of telephone calls between central offices and/or between a central office and the mobility controllers. Also, each central office has a Common Channel Inter-office Signalling (CCIS) type data link 125 going to a Signalling Transfer Point (STP) 142. CCIS type data links 140 and 174 provide data communication for PCS and related special service processing between the MC's 138, 170 and the STP 142. Also, a CCIS packet switched data link 144 connects the STP 142 to an Integrated Serves Control Point (ISCP) 146.

Each MC connects to antennas for a number of cell cites to provide **wireless** communication services to PCS portable handsets and/or other **wireless mobile** communication devices including MAPOD 2 discussed in detail below. In the example shown, **Cellular** MC 138 controls communications via a number of macrocells 140. PCS MC 170 controls communications via a number of microcells 172. The MC's 138, 170 are also interconnected with each other by IS-41 data trunks 168, and may be interconnected via voice trunks (not separately shown) essentially running in parallel with the IS-41 trunks 168.

MAPOD 2 interfaces with **cellular** mobility controllers 138 and 170 for ordering and receiving audio programming from an application provider. **Cellular** mobility controllers 138 is connected to audio/picture provider network 152 via IS-41 data trunk line 150. In addition, **cellular** mobility controller 170 is connected to audio/picture provider network 152 via IS-41 data trunk 176, switch 160 and IS-41 data trunk line 164. Alternatively, mobility controller 170 may be directly connected to audio/picture provider network 152. Audio/picture provider network 152 may also be connected to STP 142 via CCIS type data link 148 to permit some limited control exercised by ISCP 146. Audio/picture provider network 152 retrieves the audio selection from the appropriate application provider 154 and program provider 156a, 156b.

Additionally, to provide land line type centrex services for a business customer, the switch 160 provides a land line connection 178 to the customer's premises 182. The land line link would actually include a number of telephone lines connected to various types of conventional telephone terminal devices. To provide **wireless** centrex services to a particular location, which may be the

same customer premises 182, lines 180 connect the PCS MC 170 to macrocell antennae within the customer's building. Although shown as a single building, the integrated Centrex could cover a broader area, for example an entire college campus.

FIG. 9 is a diagram of another embodiment of the **mobile** audio program selection system of the present invention when used in conjunction with the MAPOD for receiving, for example, broadcast, message, data, voice, audio, and/or image information and storing same thereon as described above in detail. In FIG. 9, **cellular** system equipment 202 is constructed of conventional **cellular** equipment which can be purchased from AT&T, for example, and is used to establish **cellular** service with a **mobile** telephone by connecting a **mobile** telephone to, for example, a public switching telephone network. A main cell page monitor 204 monitors the overhead control channels to determine whether the main **cellular** system has transmitted a page message to a **mobile** telephone which may be located in the off-load **cellular** system.

If the main cell page monitor 204 receives a page message from the main **cellular** system, the main cell page monitor 204 notifies the controller 206 that a page message has been received and transmits the page message to the controller 206. The controller 206 formats the page message to be accepted by **cellular** system equipment 202 for rebroadcasting within the off-load **cellular** system. This rebroadcasting of the page message permits a **mobile** telephone to receive a page message from the main **cellular** system even while located within the off-load **cellular** system.

Rebroadcasting the page message is necessary since standard **mobile** telephones will naturally tune to the frequency or channel having the highest signal strength within a **cellular** service system or provider. Thus, with this configuration, a **mobile** telephone located within an off-load **cellular** system will be able to receive pages, voice, data, image, audio information, and the like, from the main **cellular** system to permit the main **cellular** system to access the **mobile** telephone and to be stored thereon while the off-load **cellular** system appears transparent to the main **cellular** system.

In addition to the rebroadcasting of the page message to permit the **mobile** telephone to receive pages from the main **cellular** system, controller 206 may also be used to determine whether a **mobile** telephone which is located in the off-load **cellular** system should be serviced by the off-load **cellular** system or whether the **mobile** telephone should be selectively shed from the off-load **cellular** system to receive **cellular** service from the main **cellular** system.

The **mobile** telephone's access attempt is shed by broadcasting a message instructing the **mobile** telephone to tune to the main **cellular** system. As will be discussed, the message preferably conforms to EIA-553 interface

specifications and includes a directed retry message as discussed in these specifications.

FIGS. 10A and 10B together comprise a block diagram of the present invention according to the first embodiment. In FIGS. 10A and 10B, receive antenna system 220 is used to receive signal transmissions from the **mobile** telephone. The receive antenna 220 may, for example, simply be a leaky coaxial cable connected to a 3-DB **mobile** antenna mounted in various floors of an office building. Once the receive antenna system 220 has received a signal from the **mobile** telephone, the signal is transmitted to conventional filter 222 via, for example, a coaxial cable which may be connected to filter 222 using an N-type connecter.

Filter 222 is preferably a band pass filter which limits the band width of the receive signal of receive antenna system 220. Filter 222, therefore, is used to limit the frequencies of channels which are to be considered by the off-load **cellular** system, i.e., filter 222 excludes signals which are not of interest to the off-load **cellular** system but which may be of interest to other systems, such as other **cellular** systems or marine based systems, etc. Filter 222 then transmits the filtered signal to the low noise amplifier (LNA)/splitter 224 via, for example, a coaxial cable which may be connected to the low noise amplifier/splitter 224 via an N-type connector.

Low noise amplifier/splitter 224 is conventional and amplifies the signal received from filter 222 and splits the signal into various identical signals which are then output to each transceiver 226. The signal is amplified in the low noise amplifier/splitter 224 since there is a great deal of loss in the signal. when the signal is split. The low noise amplifier/splitter 224 is connected to the transceiver 226 via, for example, a coaxial cable using, for example, an N-type coaxial connection.

Conventional transceiver 226 receives the signal from the low noise amplifier 224 which is in the standard interface format used between **mobile** telephones and **cellular** systems i.e., Electronic Industries Association (EIA)-553 publication. The transceiver 226 boosts the received signal using a preamplifier and then demodulates the signal into 10 kHz Manchester encoded data. The transceivers 226 may be preprogrammed by the transceiver control and interface system 228 to receive specific channels of interest which are broadcast by the **mobile** telephone for call registration, call origination or page response messages. The transceiver is connected to the transceiver control and interface system 228 using, for example, a 25 conductor cable assembly with a D-sub connector.

The transceiver control and interface system 228 receives the Manchester encoded data from the transceiver 226, decodes the Manchester encoded data

received from transceiver 226 and extracts information received from the **mobile** telephone such as **mobile** identifier, electronic serial number, telephone number, etc. The transceiver control and interface system 228 then sends the decoded data to the system control computer 230 using, for example, a conventional RS-232 interface cable connection. The protocol used between the transceiver control and interface system 228 and the system control computer 230 can be any standard protocol such as an asynchronous 8 bit transmission protocol. The data which is received by the system control computer 230 initially transmitted from the **mobile** telephone is typically either a **mobile** telephone registration, origination or page response message.

The operations which are performed by the system control computer 230 for the various messages are shown in FIGS. 11-16. When the **mobile** access attempt from the **mobile** telephone is a registration message, the system control computer 230 in step S2 of the FIG. 11 determines that the **mobile** registration process should be performed in step S4 which is shown in FIG. 12. The **mobile** registration process is then started in step S6 of FIG. 12 by the system control computer 230. The off-load **cellular** system will permit **mobile** registration, when the off-load **cellular** system is configured for autonomous registration, in the system control computer 230.

Autonomous registration is typically used in **cellular** systems to permit the **cellular** system to verify that a **mobile** telephone user may be provided with **cellular** service before the **mobile** telephone user has dialed a calling number and pressed a send key on the **mobile** telephone. Thus, autonomous registration permits the **mobile** telephone to be immediately connected with the calling party when dialing a calling number since the **mobile** telephone has been previously validated. When autonomous registration is not used, the **mobile** telephone placing the call must be validated, which requires additional time before the **mobile** telephone is connected with the calling party.

The registration process of the present invention utilizes the conventional registration process which detects the presence of **mobile** telephones prior to a call attempt and which is described in the **cellular** radio telecommunications system operations interface specification EIA-553. In order for the **mobile** telephone to perform autonomous registration, the system control computer 230 pre-programs the transceiver control and interface system 228 to transmit to the **mobile** telephone the standard interface message including registration bits which are set to indicate to the **mobile** telephone upon examination of the registration bits to perform autonomous registration according to the interface specifications EIA-553.

In addition, a **mobile** telephone may be validated, for example, for credit worthiness, using a conventional visitor location register (VLR) 232 which would be connected to the system control computer 230 via an RS-232 data

interface; the visitor location register 232 may then interface with a conventional home location register (HLR) service using an IS-41 **cellular** signalling network or alternatively, the visitor location register 232 could access directly via, for example, a dial-up modem, a clearing house, such as GTE Telecommunication Services, which can validate the **mobile** telephone for the off-load **cellular** system.

As shown in FIG. 12, the system control computer 230 determines whether the **mobile** telephone (or "**mobile**") is already registered in step S8, and if the **mobile** telephone is already registered, the system control computer 230 resets the de-registration counter of the de-registration process in step S10. The de-registration process constantly monitors whether the **mobile** telephone is attempting to register with the off-load **cellular** system. As shown in FIG. 16, the system control computer 230 decrements the de-registration counter in step P2 based upon a predetermined time interval. In step P4, the system control computer 230 determines whether the de-registration counter is equal to 0, indicating that the **mobile** telephone has failed to register again within the prescribed amount of time. If the de-registration counter in step P4 is not 0, then control is directed back to step P2 for decrementing the de-registration counter at the next specified time interval. If, however, the de-registration counter is 0, then the system control computer 230 deletes the **mobile** telephone from a registration customer list maintained by the system control computer 230 thus indicating that the **mobile** telephone is now no longer registered with the off-load **cellular** system. From step S10 in FIG. 12, the **mobile** registration process and the **mobile** access attempt process are then exited until the system control computer 230 receives another access attempt from the transceiver control and interface system 228.

If the system control computer 230 determines in step S8 of FIG. 12 that the **mobile** telephone is not already registered, then the system control computer 230 increments the registration message attempt counter in step S12 and determines whether the registration message attempt counter is equal to a preset data base value in step S14. If the registration message attempt counter is equal to the database value, then the **mobile** telephone has attempted to register several times indicating that the **mobile** telephone is a suitable off-load **cellular** system user and, therefore, the system control computer 230 in step S16 registers the **mobile** telephone, and resets the de-registration counter. The **mobile** registration process is then exited until the next access attempt is received by the system control computer 230.

If the system control computer 230 determines in step S18 of FIG. 11 that the **mobile** access attempt is an origination message, the origination message process is started in step S22 in FIG. 13 by the system control computer 230. The system control computer 230 first determines in step S24 whether the **mobile** identifier received from the transceiver control and interface system 228 is

included in a conventional system control computer database (not shown). If the **mobile** identifier of the **mobile** telephone which has initiated the origination message is in the system control computer database, the system control computer 230 resets the de-registration counter in step S26 and then attempts to route the call using the off-load **cellular** trunks in step S28 which is shown in FIG. 15.

The attempt to route out the **mobile** telephone call to the off-load **cellular** trunks starts in step S30 of FIG. 15. The system control computer 230 first determines whether the number dialed by the **mobile** telephone will route to the off-load **cellular** trunks in step S32. If the dialed number is not valid for the off-load **cellular** system, then the system control computer 230 will retrieve a list of frequencies to be used to remove or shed the **mobile** telephone from the off-load **cellular** system back to the main **cellular** system in step S34. These frequencies are previously determined to be compatible with the main **cellular** system, and therefore, the **mobile** telephone should be able to obtain **cellular** service using the main **cellular** system once the system control computer 230 indicates, to the transceiver control and interface system 228, to broadcast a control message preferably including a directed retry message in step S36.

The directed retry message is formulated according to existing EIA-553 interface specifications and indicates, to the **mobile** telephone, to tune to the specific frequencies of the main **cellular** system which are included in the directed retry message. After step S36 is performed, the attempt to route call process is exited until another origination message is received which includes a **mobile** identifier which matches the **mobile** identifier stored in the system control computer database.

If the dialed number will route to the off-load **cellular** trunks in step S32, the system control computer 230 determines whether the **mobile** telephone's toll class will allow the dialed number. If the **mobile** telephone's toll class will not allow the dialed number, then the **mobile** telephone is shed from the off-load **cellular** system in steps S34 and S36 by the system control computer 230 as described previously. If the **mobile** telephone's toll class will allow the dialed number, then the off-load **cellular** system will route the dialed number to off-load trunks in step S40 in order to connect the **mobile** telephone call to the telephone equipment associated with the dialed number. The attempt to route call process then exits.

If the system control computer 230 determines that the **mobile** identifier is not in the system control computer data base in step S24 of FIG. 13, an origination message attempt counter is incremented in step S42 and the system control computer 230 then determines whether the origination message attempt counter is equal to a predetermined database value in step S44. If the

origination message attempt counter is equal to the database value, then the **mobile** is registered and the de-registration counter is reset by the system control computer 230 in step S46.

The **mobile** is then shed from the off-load **cellular** system in steps S48 and S50 as previously described with respect to steps S34 and S36 shown in FIG. 15. If the origination message attempt counter is not equal to the database value, then the **mobile** is not registered and the **mobile** is then shed from the off-load **cellular** system in steps S48 and S50. After step S50 is performed, the origination message process is then exited until the next origination message is received, as discussed above.

If the system control computer 230 determines that a page response message is received from the **mobile** telephone in step S52, of FIG. 11 which is in response to a page message from either the main or off-load **cellular** systems, then the **mobile** page response process is performed in step S54. The page response message process starts in step S56 of FIG. 14, and the system control computer 230 determines whether the page response which has been received from the transceiver control and interface system 228 is a result of a page message broadcast from the main cell system. If the page response is a result of a page message broadcast from the main cell system, the system control computer 230 retrieves a list of frequencies in step S60 to include in the directed retry message which is then sent to the **mobile** telephone via the transceiver control and interface system 228 in step S62.

A list of frequencies which are used for broadcasting the directed retry message may be obtained beforehand and stored in the system control computer based upon the off-load **cellular** system's location or ability to receive the control messages from the neighboring main **cellular** systems. For example, the system control computer 230 may store only two or three frequencies to use for broadcasting the directed retry message which represent the two or three cell sites of the main **cellular** system which are located near the off-load **cellular** system. Thus, these two or three main **cellular** systems will typically broadcast the strongest signal strength for the off-load **cellular** system region and, therefore, the **mobile** telephone needs only to be informed to retune to these two or three frequencies when the off-load **cellular** system sheds the **mobile** telephone.

Alternatively, the signal strength of the various main cell systems can be monitored by the main cell page monitor 34, discussed below, and the main cell page monitor 34 can inform the system control computer 230 of the frequencies which are the strongest to insure that the **mobile** telephone tunes to a frequency of the main **cellular** system which would be of the strongest signal strength thereby obtaining better reception for voice communication.

If the system control computer 230 determines that the page response is not a result of a page message broadcast from the main **cellular** system in step S58, then the system control computer sends a message via the transceiver control and interface circuit 228 to indicate to the **mobile** telephone to use a voice circuit of the off-load **cellular** system.

If the system control computer 230 determines that the page response is a result of a page message broadcast from the main **cellular** system in step S58, the main cell page monitor 34 receives the main cell page and rebroadcasts the page so that the **mobile** telephone located in the off-load **cellular** system will receive the main cell page. The system control computer 230 which implements the above processes may be a conventional IBM compatible personal computer using, for example, a conventional 386 type microprocessor chip.

The call delivery process is a standard function of the IS-41 interface specifications which allows a **cellular** system which receives a call attempt to one of its **mobile** telephones to deliver the call to another **cellular** system which is providing **cellular** service to the **mobile** telephone using the standard IS-41 interface protocol. Once the voice circuit is connected to the incoming trunk circuit in step S64, the page response message process ends.

The main cell page monitor 234 in FIG. 10A is used for receiving page messages of the main **cellular** system via receive antennas 236 and for indicating to the system control computer 230 via a standard RS-232 data interface the content of the page messages received from the main **cellular** system. The system control computer 230 will then transmit the page message to be rebroadcast to transceiver control and interface system 228. Transceiver control and interface system 228 then formats the message as described above to enable the **mobile** telephone to receive the page message in the proper format according to EIA-553 specifications. The formatted message is then transmitted to transceivers 226 for transmitting the message to the **mobile** telephone.

The signals which are to be transmitted may be amplified in conventional power amplifiers 238 and combined in conventional combiner 240. Alternatively, for low power implementation of the above off-load **cellular** system, power amplifiers 238 are not necessary. The combined signal is then filtered using conventional band pass filter 242 and then transmitted via conventional transmit antennas 244. Thus, a **mobile** which is currently locked on to the frequencies in the off-load **cellular** system will still be able to receive its pages from the main **cellular** system.

Once the **mobile** is registered (steps S16 or S46) via IS-41 procedures, the main **cellular** system may provide busy features such as indicating to a telephone equipment user trying to reach the **mobile** telephone that the **mobile** telephone is currently in the busy status as well as providing call forwarding

and call waiting features when the **mobile** telephone is provided with **cellular** service from the off-load **cellular** system. In addition, the **mobile** telephone which is locked on to the off-load **cellular** system will be able to receive similar call treatment as the main **cellular** system such as specific dialing patterns, restrictions, activated features, etc. Additional details of the off-load **cellular** system or micro-**cellular** system are disclosed in U.S. Pat. No. 5,487101, incorporated herein by reference.

FIG. 17 is a block diagram of a standard ADSL arrangement. To preserve POTS and to prevent a fault in the ADSL equipment 354, 356 from compromising analog voice traffic 366, the voice part of the spectrum (the lowest 4 kHz) is separated from the rest by a passive filter, called a POTS splitter 358, 360. The rest of the available bandwidth--from about 10 kHz to 1 MHz--carries data at rates up to 6 bits per second for every hertz of bandwidth from data equipment 362, 364. The ADSL equipment 356 then has access to a number of destinations including significantly the Internet 368, and other destinations 370, 372.

To exploit the higher frequencies, ADSL makes use of advanced modulation techniques, of which the best known is the discrete multitone (DMT) technology. DMT was pioneered by Stanford University, California, and Amati Communications Corp., San Jose, Calif., and endorsed by the American National Standards Institute (ANSI), New York City, and the European Telecommunications Standards Institute (ETSI), Sophia Antipolis, France.

DMT divides the bandwidth from about 10 kHz into a set of 265 independent subchannels, each 4 kHz wide. By measuring the quality of the subchannels and then assigning a bit-rate to each based on its quality, DMT customizes the transmit signal for every line. In doing so, it automatically avoids regions of the frequency spectrum that are too noisy or too attenuated to support reliable communications. If the quality of a subchannel degrades enough to affect a system's error performance, the data rate on that subchannel is lowered and the excess traffic moves to a subchannel capable of supporting it. The result is robust communications over single twisted pairs.

As its name implies, ADSL transmits data asymmetrically--at different rates upstream toward the central office 352 and downstream toward the subscriber 350. Such a technology makes sense for two practical reasons. For one, the typical WWW surfer is more interested in downloading large files than in uploading them, and therefore needs more capacity in the downstream (network-to-subscriber) direction.

The second reason is technical: when many wire pairs are squeezed together in a cable, cross talk is inevitable. Signals traveling downstream from the central office 352 are not much affected, because they are all of approximately

the same amplitude. On the other hand, upstream traffic originates in subscriber premises 350, and these buildings may be at different distances from the points at which lines come together in a cable; accordingly, upstream signals can vary greatly in amplitude. If a wire pair carrying a strong signal shares a cable with another wire pair carrying a weak one, cross talk can be all too evident. But since cross talk increases with frequency, the problem can be made tractable by limiting the upstream data rate and keeping it near the low-frequency end of the spectrum.

Meanwhile, cable television providers are not sitting by idly. They want to provide Internet service to PC users over their TV cable systems by means of special cable modems. Such modems are capable of transmitting up to 30 Mb/s over hybrid fiber/coax systems (which use fiber to bring signals to a neighborhood and coax to distribute it to individual subscribers. Further, they are available, and they work.

Cable modems come in many forms. Most create a downstream data stream out of one of the 6-MHz TV channels that occupy spectrum above 50 MHz (and more likely 550 MHz) and carve an upstream channel out of the 5-50-MHz band, which is currently unused. Using 64-state quadrature amplitude modulation (64 QAM), a downstream channel can realistically transmit about 30 Mb/s (the oft-quoted lower speed of 10 Mb/s refers to PC rates associated with Ethernet connections). Upstream rates differ considerably from vendor to vendor, but good hybrid fiber/coax systems can deliver upstream speeds of a few megabits per second. Thus, like ADSL, cable modems transmit much more information downstream than upstream.

The downstream channel is continuous, but, like Ethernet, divided into packets, with addresses in each packet indicating for which subscriber each is intended. the upstream channel has a media access control that slots user packets or cells into a single channel.

To avoid collisions, in some cable systems, upstream packets are gated onto the network via control signals embedded in the downstream information. Other approaches divide the upstream path into frequency channels and allocate a channel to each user. Still others combine these two multiplexing methods. A few modem companies are proposing techniques like spectrum spreading or code-division multiplexing to reduce susceptibility to interference from antennas and other sources of electromagnetic radiation outside the system. Called ingress noise, it is the biggest difficulty on hybrid fiber/coax networks.

Variation in the capacity of cable systems depends less on cable length than on ingress noise and on the number of users seeking simultaneous access to a shared line. (Cable data rates are not particularly sensitive to the length of

the coaxial cable; amplifiers in the cable network keep signal power high enough to make length a minor consideration.)

Because cable TV systems use a shared-bus architecture, they may be less expensive to implement than ADSL. But that shared architecture is a double-edged sword. As with any shared medium, as more users go on-line, the capacity available to any one user inevitably falls.

At present, the point is somewhat academic since the top speeds of both ADSL and cable systems will not be usable for years anyway. Internet server speeds, network delays, and personal computer limitations will hold usable rates at or below 2 Mb/s for the foreseeable future. So far, ADSL offers higher security and reliability. Cable modems may offer a less expensive network solution because of the cable plant's shared architecture, but that differential is more than offset by infrastructure costs required to upgrade existing coaxial cable networks to hybrid fiber/coax. The technologies for both ADSL and cable modems are at about the same state of maturity and integration.

ADSL's greatest advantage is that it can make use of existing twisted copper pairs, which are numerous indeed compared with the number of hybrid fiver/coax lines that exist in upgraded cable systems. Today the global ratio is on the order of 600 million to 6 million, or about 100:1. In the United Sates, it is about 20:1. Even with aggressive cable upgrades, the numbers are not likely to reach parity over the next five or six years. Additional details regarding the above communication trends can be found in "Communications," IEEE Spectrum, p.27, January 1997, incorporated herein by reference.

We have also realized that there is yet another communication network and network design emerging. That is, to relieve the problem of poor performance for the Internet, about 100 U.S. researcher universities have joined forces to develop an ultrafast Internet--Internet 2 (see htpp://www.internet2.edu, incorporated herein by reference). However, Internet 2 is not designed just to create a fast network. Instead, it will also let researchers design the types of applications that could be used on fast networks.

Many universities already have network connections that will let them participate in Internet 2. Meanwhile, the North Carolina Giganet is already operating with Internet 2 architecture. This ultrafast network serves Duke University, North Carolina State University, and the University of North Carolina, Chapel Hill.

Internet 2 is decentralized. The participating institutions will decide many issues for themselves, such as the way they will connect to Internet 2 as the way they will connect to Internet 2 and who at their institutions will have access to the system. Some of the applications that will be developed on

internet 2 may be ready by October 1997, although the program's target date for having applications online is October 1998.

Developers expect that Internet 2 will operate at 2.5 Gbps. Most of the current Internet runs at 45 Mbps, although some privately operated segments run at 155 Mbps. Developers will build Internet 2 on existing equipment and networks as much as possible. The single most expensive element, the core fiber-optic backbone, already exists as vBNS, the very-high-speed Backbone Network Service.

vBNS currently serves the five U.S. supercomputer centers and several universities that have recently connected to it. vBNS uses ATM networking over Sonet (the high-speed, fiber-optic switched synchronous optical network). Sonet uses gallium arsenide microelectronics to achieve high-speed, heavily loaded switching. vBNS originally was capable of 155 Mbps but since February has been capable of 2.5 Gbps.

As shown in FIG. 18, Internet 2's 380 major nodes will be secure, specialized, high-speed network connection points called gigapops 374, 376, 378 (gigabit-capacity points of presence). The gigapops will provide all the equipment necessary to connect a set of universities 382, 384, 386, 388, 390, 392 to the vBNS backbone.

The set of users who hook up to each gigapop will determine exactly what form it will take, including the type of equipment it will use. Initially, the gigapops will be connected to each other by vBNS, through which they will receive fast network services. However, Internet 2 participants may develop their own central connection system in several years.

The basic network line level, ATM, will permit the broadband communication of everything from multimedia applications to TCP/IP applications. Internet 2 will use RSVP (resource reservation protocol) to manage the quality of service of real-time, data-intensive multimedia applications. At the network layer level, Internet 2 will support the current Internet Protocol version 4 and IPv6 (IP Next Generation), which is still under development. In fact, Internet 2 will be the testbed for many IPv6 concepts. Internet 2's design focuses heavily on maintaining predictable and dependable broadband, high-speed throughput by strictly controlling who uses the system, what they may use it for, and how they transmit data. For additional discussion on Internet 2, see "Tomorrow's Internet is Here Today," IEEE COMPUTER, p.22, April 1997, incorporated herein by reference.

FIG. 19 is an illustration of the architecture of the combined internet, internet 2, POTS, and ADSL architecture. The internet 2 architecture 380 and ADSL architecture 354, 356 is combined with the standard internet architecture

400 with user networks 398, 402, and 404. Advantageously, **mobile** switching office 14 and/or 20 (asymmetric or symmetric as discussed above in detail) services or interfaces with MAPOD 2, 2a, 2b, 2c and/or 2d to transmit to, and/or receive from, data, **voice mail** messages, electronic mail messages, voice, audio and various other communications as discussed above in detail.

Advantageously, MAPOD 2, 2a, 2b, 2c and/or 2d is capable of receiving various different types of messages including, for example, messages transmitted from a customer ADSL 350, internet 400, internet 2 300, telephone 396, and the like. As illustrated in FIG. 19, internet access point 368 may advantageously be connected to telephone company network 370 to forward messages received from, or transmit messages to, internet related providers 400, 380. Further, telephone company network 370 is also advantageously connected to one or more ADSL access modules 356 described above. As described below in greater detail, the present invention may also advantageously be used in the context of uploading a data, **voice mail** and/or electronic mail message to be transmitted to another user, **mobile** and/or land based, either terrestrial, **mobile** user, internet based and/or ADSL based user.

FIGS. 20-23 are flowcharts describing in detail the process flow of the **mobile** telephone receiving an incoming message comprising one or more of a voice message, a data message, an electronic mail message, an internet message, and/or and ADSL message for storage on the handset, and also for subsequent uploading to another destination. In FIGS. 20-23, the **mobile** switching office receives an incoming message from an external source and/or data provider in Step T2. The incoming message may be, for example, an electronic mail message, a voice message, an audio message, a data message and/or an image message. The external source and/or data provider may be transmitted via, for example, the internet, public switched telephone network (PSTN), **mobile** network, audio and/or video provider, and the like.

The **mobile** switching office optionally determines whether the **mobile** is in the service area to be able to receive the transmission from the **mobile** switching office in Step T4. In Step T6, if the **mobile** is determined not to be in the service area, the **mobile** switching office optionally informs the external source or provider that it was unable to currently make the connection in Step T8. If the **mobile** switching office does not optionally make a later attempt in Step S10, then the process ends in Step T12.

If the **mobile** switching office does optionally make a later attempt, then the **mobile** switching office waits a predetermined period of time in Step T14, and broadcasts or transmits the message to the **mobile** in Step T16. If in Step T6 the **mobile** is determined to be in the service area, then the **mobile** switching office broadcasts or transmits the message to the **mobile** in Step T16.

The **mobile** switching office next optionally checks for the **mobile** response in Step T18. If the **mobile** switching office does not receive the response from the **mobile** in Step T20, then control of the process is reverted to Step T10. If the **mobile** switching office receives the response from the **mobile** in Step T20, then the **mobile** stores the incoming message in a memory on the handset in Step T22, and notifies the user of the **mobile** of the incoming message in Step T24, via, for example, audible, visual and/or vibration means. As discussed above, the incoming message may be, for example, an electronic mail message, a voice message, an audio message, a data message and/or an image message, or any other message.

In Step T26, the **mobile** telephone determines whether the user has responded, and if not, the **mobile** waits a predetermined period of time and optionally re-notifies the user of the **mobile** in Step T28. The notify count is optionally incremented in Step T30, and the **mobile** determines whether the notify count has exceeded a predetermined threshold in Step T32. If the notify count has not exceeded the predetermined threshold in Step T32, then control is reverted back to Step T28.

If the notify count has exceeded the predetermined threshold in Step T32, then the **mobile** optionally activates a less obtrusive notification process such as, for example, only vibration, a lower audible sound, and the like in Step T34. The process then ends in Step T51.

If in Step T26 the **mobile** telephone determines that the user has responded, the **mobile** provides the user with various message access features in Step T36. The various message access features include, for example, listening to the **voice mail, e-mail,** or other mail message; reading on the display of the **mobile** telephone one or more of the above messages; deleting the messages; saving the messages; date stamping the messages; and the like.

The user then enters or inputs one or more of the various access commands in Step T38 as described above. The user is then presented with the option of sending and/or forwarding and/or broadcasting the received messages in Step T40. The **mobile** then determines whether the user has indicated that the message is to be forwarded and/or broadcast and/or sent in Step T42. If the user indicates that the feature is not to be used in Step T42, then the process ends in Step T51.

If the user indicates that the feature is to be used in Step T42, then the **mobile** optionally formats and/or optionally encodes and/or optionally **compresses** the message for uploading to the **mobile** switching office in Step T44. In Step T46 it is determined by, for example, the **mobile** switching office, whether the message transmitted from the **mobile** is to be routed to a PSTN, and if so, the **mobile** switching office transmits the message from the

mobile to the PSTN in Step T48.

The **mobile** switching office optionally determines, with the optional assistance of the PSTN, whether the message upload was successful in Step T50, and if so, the process ends in Step T51. If the message upload was not successful in Step T50, then the **mobile** switching office optionally determines whether there has been exceeded the predetermined number of attempts in Step T52, and if so, the process ends in Step T51.

If the **mobile** switching office optionally determines that there has not been exceeded the predetermined number of attempts in Step T52, then the message attempt counter in incremented in Step T54, and the process returns to Step T48.

If it is determined in Step T46 by, for example, the **mobile** switching office, that the message transmitted from the **mobile** is not to be routed to a PSTN, then it is determined by the **mobile** switching office whether the message is to be routed to another **mobile** in Step T56. If the message is to be routed to another **mobile** in Step T56, then the **mobile** switching office performs the upload operation to upload the message to another **mobile** telephone in Step T58. The **mobile** switching office optionally determines whether the upload was successful in Step T60, and if so, the process ends in Step T51.

If the **mobile** switching office optionally determines that the upload was not successful in Step T60, then the **mobile** switching office optionally determines whether the maximum number of attempts has been exceeded in Step T62, and if so, the process ends in Step T51. If the maximum number of attempts has not been exceeded in Step T62, then the message attempt counter is incremented in Step T64, and the process reverts to Step T58.

If it is determined in Step T56 by, for example, the **mobile** switching office, that the message transmitted from the **mobile** is not to be routed to another **mobile,** then it is determined by the **mobile** switching office whether the message is to be routed to an ADSL address in Step T66. If the message is to be routed to an ADSL address in Step T66, then the **mobile** switching office performs the upload operation to upload the message to an ADSL address in Step T68. The **mobile** switching office optionally determines whether the upload was successful in Step T70, and if so, the process ends in Step T51.

If the **mobile** switching office optionally determines that the upload was not successful in Step T70, then the **mobile** switching office optionally determines whether the maximum number of attempts has been exceeded in Step T72, and if so, the process ends in Step T51. If the maximum number of attempts has not been exceeded in Step T72, then the message attempt counter is incremented in Step T74, and the process reverts to Step T68.